ATARI COMPUTER ENTHUSIASTS

3662 Vine Maple Dr. Eugene OR 97405

NOVEMBER, 1984

Editors: Mike Dunn, Jim Bumpas, Larry Gold

News and Reviews

by Mike Dunn, Co-Editor

We have received a number of phone calls from Atari explaining they are trying to produce the 800XL's and provide customer support. They are beginning with a new magazine, called **The Atari Explorer**, edited by Neil Harris, formerly an editor of a similar Commodore magazine. Mr. Harris appears to be very interested in customer support as well as User Group support. Each issue will feature a User Group. The first will be ours since we were the first! They want to write about both large and small groups and show how they help Atari owners. At the present time, the entire customer support group consists of only 6 people, so Atari is very interested in supporting user groups all over in providing help such as our **BugBusters** we are trying to form. We want to have a list of people willing to help others by phone or mail. We'll put the list in the Newsletter with the hours you're available to be called. This will be for only as long as you wish, and hopefully we of ACE as well as other groups all over the country could set up a network to take the place of the support system formerly supplied by Atari.

Many new items are going to be available soon for the Atari 800XL. Rumor tells us our good friends at **MicroBits** in Albany, OR will be coming out with a 1200 BAUD modem as well as a hard disk with an OS by Bill Wilkinson of OSS. These will be sold at their usually unbelievably low prices! **AXLON** will be releasing the fabled expansion interface box for the 800XL and **Batteries Included** will add an 80 column board. I don't know anything about any future Atari computers, but again doubt if they will have any compatibility with our 800's. However for the price (now $170 and dropping), you can not beat an Atari system.

There are new products coming out for the Atari now. **Synapase** has lowered the prices on all their products now if you get them direct from their office at 5521 Central Ave., Richmond, CA 94804. They even have very low prices on bulk orders for user group members. **The Scarborough System** 25 N. Broadway, Tarrytown, NY 10591, has released a new improved **MasterType** for $40. They have a very nice line of programs, including the fabulous **Your Net Worth** (see below). AMDEK is selling to user groups their 3" disk drives very cheaply. One of the members of the Portland Atari Group has been able to get hold of a large number of brand new Osbourne Executive computers, CP/M based portables with two disk drives, 7" Amber screen, 128K etc for only $895 with a ton of software. I bought one to go with my ATR 8000 CP/M system. If anyone is interested, call Jim at 503-646-3467 for details.

This month, we introduce a new regular contributor, Ralph Walden. Ralph will be writing a column for advanced users in Assembly language and C. He has modified the Deep Blue C from APEX, and the original author will allow ACE to put it our library for members! Not quite ready, but expect a lot on C in the future. Ralph likes C so much because he can use the same program for various computers, including big mainframes. He was also the first volunteer for the BugBusters for help in C and assembly language.

Reviews

by Mike Dunn

Your Personal NetWorth (Scarborough System, 25 N. Broadway, Tarrytown, NY 10591, $80)

Brought to you by the people who produced MasterType and other fine programs for a variety of computers comes this very comprehensive, quick and easy to use home accounting system. When the package arrives the first thing noticed is the very attractive presentation of their produuct. It comes in a nice plastic box holding the manual, two disks, and in front, a real silver dollar, so if you spend your last penny on the package, you still have $1 worth of assets. The manual is a model for the industry in its clarity, not only about using the program, but the explanation of accounting in general. It explains assets and liabilities, as well as when and why to use each. Also included in the package is the book by Silva Porter on money management.

After booting in the program disk, you begin by establishing your accounts. You have a choice of many accounts to use. There are 32 expense accounts, 30 asset accounts, 16 liability accounts and 16 income accounts, plus you can add your own. You then begin by entering your initial balances for asset and liabilities. You can make entries, post entries in a journal of accounts, make budgets, keep bank records, make inquiries, and print out your complete balance statement anytime you want. You can have as many as 10 bank accounts, 350 different account categories, and up to 4000 transactions per disk. Other features include household inventory, stock portfolio analysis, all kinds of help features and trace features, easy bank statement reconciliation, and many other features too numerous to mention.

Not only is the manual outstanding, there are many help screens to guide you in any operation. The screens are attractive and easy to use; the special features of the Atari are well utilized, a surprising feature in a program available for many computers. The program is written in machine language and all operations are fast! If you need such a program, plan on spending several hours entering all your original accounts and data, then it is very easy to keep it up. A perfect program to get for the New Year, and then next year, to get your complete balance sheet, as well as all your income tax information, just press a button!!

The XL Boss (Allen Macroware, POB 2205, Redondo Beach, CA 90278, $80)

From the makers of DiskWiz and PrintWiz comes Jerry Allen's first hardware project. This is a chip for the 800XL or 1200XL (different one for each) — an OS replacement. It allows you to run almost all of the Atari Software without having to use the translator disk. Other features include not having to hold down the OPTION key to de-select BASIC, the ability to select the "hidden" 4k of RAM for word-processors, spreadsheets, etc., and, most interesting, the ability to have a RAM rather than ROM operating system. You can then easily customize your operating system as you wish, without having to burn an EPROM, etc. You can also use this feature to load in Jerry's monitor program **MacroMon XL**, then using the warmboot command, boot in any disk, switch back and forth from the monitor to the program, and see how they do various tasks on commercial boot programs. This monitor, besides the usual functions, has split screen capabilities to compare two parts of memory, has a find command can read and write directly to the disk.

The manual gives you ideas on how to modify your OS, such as changing the cursor arrow keys to work without the shift key. As Jerry always has done, the chip will be updated occasionally; the first one out in 2 weeks to allow you to shift back and forth from the standard OS and the Boss XL, so if anyone comes out with an expansion box, etc., you can use it.

In using the chip, I find most programs will load in and run, but some of the older commercial terminal programs didn't. The manual says to switch back and forth between the various options of the XL Boss and the Monitor, when using the function keys, to push the keys just right — so true! The keys are very touchy and need to be pressed the way Jerry states, not because of the chip, but because the keys are not the best quality and are touchy. The installation of the chip in my 800XL required taking the entire computer apart, but the total operation was easy and well explained; it took about 20 minutes. The 1200XL model is $10 more.

PrintWiz (Allen Macroware, $30.

This is an update of the program we have used for quite a while to dump the pictures seen in ACE of our programs. Present owners can send their old disk in and get the upgrade for only $5. New features include adding the DMP-80, Panasonic KXB1090, Mannesman-Tally Spirit 80 & IDS MicroPrism printers to the Epson, Nec, Prowriter and Okidata already supported, a program to directly use Atari Paint and Koala Pad files for dumping and using the LOGO cartridge with PrintWiz. Jerry will continue to update the program and provide those updates at a nominal charge to present owners.

PILOT Directory

A number of articles and programs have been written about accessing the disk without going to DOS in BASIC. One useful thing for a programmer is to look at the contents of the disk without losing your program.

Loading and saving in PILOT is a more lenghthy task than in BASIC since the type of "save" used is actually a LIST to disk. Unless you have monkey wrench or OS/A+ you can use up about two minutes going to DOS, using memory save, and returning to PILOT.

DIR is a 17 line mini program letting you LOAD and then type J:*DIR to run. It takes about two seonds to load and another few seconds to run. The total time required to access the disk is about five seconds the first time and two seconds once the program is in your Atari. The line numbers used are the last twenty usable in PILOT (9980-9999). If your disks have less than 40 files you may shorten this program by (9 lines) deleting the counter and page turner routine.

— Carl Schwartz

BUMPAS REVIEWS

The Writer's Tool (OSS, $99 until Dec. 31, a $30 discount) is a cartridge-based word processor of 20k with a disk containing utilities, demos, and configuration files for numerous printers. This is a full-function program which will meet the needs of the busiest writer. Yet I believe it is easy enough to use in its default configuration for anyone familiar with a typewriter.

You need at least a 48k system and a disk drive. Double density drives are supported. Atari, Epson, Gemini, Prowriter, NEC, Okidata and Comrex printers are directly supported by configuration files provided on the disk. There is also a "generic" printer file for all others. In addition, you may put printer control codes directly into a text file in order to use unsupported printers more fully. It usually takes 6 keystrokes to toggle these unsupported features, but with a printer like mine (IDS480) you're just happy to be able to use the features at all!

OSS informs the user of one known bug: double-wide printing is not properly supported when the line is justified. Double-wide printing works fine in any other context. I think they mean only to speak of supported printers here, because I experienced anomolous behavior in any line in which I combine double-wide with any other printing. I must say this same experience occurs with any word processor I've used for unsupported printers.

The opening screen permits the user to begin typing text immediately. A prompt at screen bottom tells you to press the Option key for the main menu. All full-function word processors should start you off this way. But OSS is one of the few to do so. OSS has provided the most extensive set of cursor controls I've ever seen on a word processor for the Atari. In addition to the usual Tab and cursor arrows, you can move to Beginning or End of file, beginning or end of a line, beginning of the next line, beginning of the next word, and beginning of the next page or the previous page. The cursor may also be moved by the search process. During a save operation to disk, if the cursor is not at the top of file, you will be warned only part of the file will be saved. If you answer "Y" to the prompt for saving all the file, the cursor will then move to top of file. You can also choose a block or underline cursor, and you can adjust the rate at which the cursor flashes.

Editing can be accomplished easily in either an insert mode or a typeover mode. In addition to the usual insert commands, you can also insert all available space (pushing all text after the cursor to the end of the 24k buffer), allowing you to type in some material and then delete all space to the next character to close up the gap. Most editing commands require only one or two keystrokes, and this is where you'll find the economy of keystrokes the most important. Two keystrokes will also "undelete" the last line deleted. This is a nice feature. Previously entered text can be converted to upper or lower case automatically with a 2-keystroke control sequence, saving you some re-typing. The search and replace function also provides a means to delete strings of text by failing to replace the string. You may also delete all text before or after the cursor.

Block operations allow moving text by copying a block to a new location, then deleting the original block. Blocks are highlighted in inverse video. A status line at screen bottom will show you when inverse video is active, and when the caps lock key is active. You are also told how many characters and words are in the buffer, and how much space remains.

Supported printers with the capability can print in the following modes: emphasized, double-wide, italics, double-strike, underline, super- or sub-script. In addition, "split justification" (dividing a line at any point, with characters to the left blocked left, those to the right blocked right), soft hyphens (not used if the word fits in a line) and hard spaces (forcing characters split by a space to be printed on the same line) are available for your use.

Headers and footers can be set to any number of lines and can contain any editing or formatting commands available in the rest of text with the addition of page numbers. The printer can be directed to stop at any point (to change print wheels, for instance).

Dynamic page breaks are not displayed in the editing mode, but on the print menu you can preview the printed text to see where the page breaks occur. Print formatting commands include all the usual page and line lengths and spacing, footer placement, centering, tabs, margins and indentations. But they also include page eject, conditional page eject (to make sure certain lines may be printed on the same page), and alternating the sides of split justified lines on even numbered pages. A page wait command is available for single sheet printing. And double-column printing is accomplished by the simple expedient of telling the program how many spaces you want between the two columns. All your other formatting commands are adjusted automatically for the second column! (This is a great feature!). Two or more files may be "linked" together for chain printing.

The program makes good use of sound to warn of errors and to note different actions. There are 20 custom error messages to lead you to the correct action after a mistake. A buzzer warns and the prompt is re-displayed upon any other error. A Basic program is included on the disk to create custom "macros" to set formats you commonly use and which differ from the defaults provided. The 174-page manual comes in a yellow vinyl 3 ring binder and describes how to incorporate graphics into the body of your printed text. An errata sheet also promises the future ability to program key sequences to translate to graphic sequences. The example they give is Control-C might be used to print copyright symbol. They are also working on developing conversion programs to use to print graphics from MicroPainter, Koala Pad, Atari Artist Pad, etc. files within your text files. If you buy the program now, you will receive the next update free of additional charge.

I like this program. The relatively high price may keep it out of most people's libraries since other full-function word processors can be purchased for as little as $50. But for someone who does a lot of writing on the Atari, this program will pay for itself with its quality.

SynTrend (Synapse Software, $70) is a two disk package for graphing, statistical analysis and forecasting. Those familiar with SynCalc and SynFile+ will recognize the pull-up menus which lead you through the program. A student of statistics, or someone using statistics a lot may find this package very helpful.

The two disks contain the programs SynStat and SynGraph. SynStat is really a spreadsheet with some partiularly special functions for use in statitical analysis. SynTrend is fully compatible with SynCalc and SynFile+, and with VisiCalc. Once the data file is created or loaded into the program, the "Analyze Data" option on the menu. The program then lists all the variables (column labels in the spreadsheet) you used. Picking one, the choice of "Descriptive" will show the number of entries in the column, their average, standard deviation, variance, standard error, minimum, maximum, and range.

The other analysis performed by SynStat is "Regression" analysis. For this mode you select one variable as a dependent variable and one or more independent variables. This mode is designed to show the relationship between variables. The screen shows calculations for coefficient, standard error, Rsquared, and AdjRsquared. The manual advises the user to refer to a text book on statistics if these terms are not familiar.

The results of regression analysis may also be shown on a screen titled "Residual Analysis". Here columns showing the actual entries, together with the predicted entries and the difference between actual and predicted (residual). Any of these screenfulls of calculations can be dumped to a printer.

SynGraph is used to create labeled, high resolution, color-coded graphs from data entered in SynGraph, SynStat, SynCalc, SynFile+, or VisiCalc. You may choose a line graph, bar graph, scatter plot, or pie chart. The bar graphs may be of the cluster or stacked types. Graphs may be rescaled and relabeled. Up to three factors at a time may be placed on the same graph. The graphs may be saved in disk files for later use, or dumped to a compatible printer.

This is a fun package to use. And for small business use, one might find it helpful for analysis of sales data. If you're a student, the printed output will make your papers and projects very professionl indeed.

Christmas Bazaar: Our December 12 meeting will include a bazaar. You may bring any software (originals with full documentation only!) and hardware you want to sell or trade. You may set up your wares on a table, or browse among the tables. This will also be a benefit for WISTEC. You may keep all the proceeds you receive except for 10%. The club will record the amount received from each person and WISTEC will give you a receipt you may use for your taxes.

Last month Stephen Warn reviewed S.T.A.R.P.A.D. and omitted price and vendor information. He sends it for this issue: DaVinci Soft, 4414 Murrayhill Road, Charlotte, NC 28209 (704) 523-2460. $35.00.

— Jim Bumpas, co-editor

CLEAR FOR ACTION

Clear for Action (Avalon Hill, $30) is a game of fighting ships during the age of sail. You have the option of playing any one of twelve different scenarios, or creating your own. In addition to creating your own scenarios, you may also design your own ships. The manual recommends you follow the general ship guide for criteria. This guide provides you with a list of standard ships of this time period and their various characteristics. As captain of one or more sailing ships you may play either the computer or another person. As this program is in Basic, playing the computer is a little slower than playing with a live opponent, but no less challenging.

The game itself is played on two screens which are alternately displayed. The first screen is a strategic level view of the situation. It shows the current sailing orders for all ships. Each ship is depicted by a number on a dot matrix grid. As the captain of a vessel you may change direction and sail configuration at this time.

The next screen is a tactical level screen. This shows each ship by a graphic display and gives its current relationship to every other ship. To one side of this grid you see a ship's outline, complete with rows of cannon. In the middle of this silhouette may be found four boxes. Each box depicts the current type of load for that broadside. After firing a broadside you are able to load with one of four types of shot. Each has different ranges and effects on your opponent's ships. It is on this page that you are given the opportunity to fire on your opponent's ship. After this you may go to a third text page, on which you may move crew sections from one of your four ship areas to another. Each of these areas controls the performance of either sailing or fighting your ship. Then you may proceed to the first screen and start the cycle over again.

I found this game to be easy to learn yet quite challenging to play. Avalon Hill has documented the game in two easily understood booklets. The first explains the basic game mechanics and allows you to proceed immediatley to play. The second booklet lists ships, scenarios and various charts depicting the probablity of such occurances as fouling, grappling, sailing speeds, crew functions, morale etc. I have only two complaints about this program. It is difficult at times to tell just where on the tactical screen a move on the strategic screen will place you. Thus, I have found myself continually getting a full broadside when I had thought I would clear my opponent's arc of fire. My second complaint, which in no way detracts from the game's playability, is the simplicity of the graphics and sound effects. One of Atari's strongest aspects is the quality of the sound and graphics which it may produce. An excellent game like this deserves the best graphics possible.

I highly recommend this game. It is one which can be played for many years and still be challenging.

— Nick Chrones

LET'S TALK CONTROL

(Reprint: The Escape — Little Rock Atari Addicts)

About once a year, whether I need to or not, I enter DOS and rename a file to a filename already existing on the disk. If you've ever done this you know what the problem is. You've got two files on the same disk, both with the same filename. If you try to change one, DOS changes both. If you try to delete one, DOS deletes both. If you try to load once, you only get one but it's the first one in the directory.

Is there a way to recover the other file? The answer is yes. The program listed below is called ALTERDIR (alter directory). It will read the disk directory directly from the disk (without the help of DOS), allow you to change any filename, and write the altered directory back out to the disk. The machine language subroutine in the program is a disk handler allowing you to do all kinds of things to disk data. In this case it is set up to read the 8 sectors of the disk directory. After you have entered your changes the POKE on Line 2070 changes the ML routine to write the directory data back to the disk.

This is not a program you'll need every day, but the day you do you'll be glad it's around.

— John David McFarland

6502 DISASSEMBLER

One thing which can really help a machine language programmer is a disassembler to print out his or somebody else's code. If you don't have one, it's not worth $30 to buy a disassembler limited to listing code from memory or disk file only. A disassembler may sound easy to make, but it isn't.

There are 56 different 6502 ML instructions, some have only one form, but most have up to four different forms performing up to four different things. Because of the number of possibilities, it isn't practical to use upwards of 120 different IF-THEN statements.

In my disassembler, rather than using ten million IF-THENs, I used a big string subscripted into 4 byte chunks holding the 3 letter ML instruction and an option byte. The option byte tells the program what form to use in the disassembly. The disassembler looks up the instruction by taking the number it is trying to disassemble, multiplying it by 4, and subtracting 3. This gives the instruction's location in the string. The rest is easy; decode the option byte, get any other bytes needed to complete the form of the instruction, and print it.

Type in the program, SAVE it and RUN it. The program will set up the string and prompt you from where to disassemble. If you choose the disk option, a disk directory will come up. Then enter a BINARY LOAD file name. Take care to see the # of sectors in the file isn't greater than the available buffer space, or when it loads, the program will probably crash. If you chose the memory option, enter the start address in hex or decimal. If you use hex, precede the address with a '$'. Next, choose where the output will go. If you choose the printer, be sure it is ready BEFORE you press P. If you type D, enter a filename and hit return. The resultant file will be text, but it will be pretty long if your source code is good sized.

When the program is disassembling, pressing any key will pause it. Then press START to exit back to the start of the program. Press SELECT to continue the listing, or press OPTION to end.

— Greg Menke

FORKLIFT

In the October issue of ANALOG magazine Tom Hudson presented a routine for printing text in graphics 8 mode (Graphics 8 Character Generator, p. 57). I've modified the routine so you can print blocks of characters using the call A=USR(ADDR,X,Y,BADR,SIZE,WIDTH) where:

ADDR is the address of the string holding the routine (see lines 4000-4100)

X is the X position as in graphics 0 (0-39)

Y is the Y position as in graphics 8 (0-152)

BADR is the address of the string holding the block of characters to be printed

SIZE is the number of characters in the block

WIDTH is the number of characters in each horizontal row of the block

I've used the routine in a program called 'Forklift'. The object is to take three rhyming words from a pile of seven on the left and stack them in a pile on the right. A joystick in port #1 controls the forklift. The words can only be placed down in either the left or right pile so a little planning ahead is required

— Stan Ockers

IN A WORD

This little quicky is a scrabble type game in LOGO. The BEGIN module sets the counters to 0 (SCORE and VALUE). The SCRABBLE module takes a letter input and gives it a value by using the LETTER module. The * is used to denote the end of the word being played. We could have used the IF (OR (:CHAR) = "A etc. in the LETTERS module, but felt that it made the program run slower. The SCORE module does the arithmetic to give the score in a word. Now we can see how much those $10.00 words we use at our house are really worth.

— Ruth Ellsworth

RUTHS pilot

```
TO BEGIN
MAKE "SCORE 0
MAKE "VALUE 0
PR [TYPE A WORD DO NOT TYPE RETURN. TYPE AN * AT THE END
OF THE WORD.]
SCRABBLE
END

TO LETTERS :CHAR
IF :CHAR = "A [MAKE "NUM 1]
IF :CHAR = "B [MAKE "NUM 3]
IF :CHAR = "C [MAKE "NUM 3]
IF :CHAR = "D [MAKE "NUM 2]
IF :CHAR = "E [MAKE "NUM 1]
IF :CHAR = "F [MAKE "NUM 4]
IF :CHAR = "G [MAKE "NUM 2]
IF :CHAR = "H [MAKE "NUM 4]
IF :CHAR = "I [MAKE "NUM 1]
IF :CHAR = "J [MAKE "NUM 8]
IF :CHAR = "K [MAKE "NUM 5]
IF :CHAR = "L [MAKE "NUM 1]
IF :CHAR = "M [MAKE "NUM 3]
IF :CHAR = "N [MAKE "NUM 1]
IF :CHAR = "O [MAKE "NUM 1]
IF :CHAR = "P [MAKE "NUM 3]
IF :CHAR = "Q [MAKE "NUM 10]
IF :CHAR = "R [MAKE "NUM 1]
IF :CHAR = "S [MAKE "NUM 1]
IF :CHAR = "T [MAKE "NUM 1]
IF :CHAR = "U [MAKE "NUM 1]
IF :CHAR = "V [MAKE "NUM 4]
IF :CHAR = "W [MAKE "NUM 4]
IF :CHAR = "X [MAKE "NUM 8]
IF :CHAR = "Y [MAKE "NUM 4]
IF :CHAR = "Z [MAKE "NUM 10]
IF :CHAR = "* [START]
END

TO SCRABBLE
MAKE "CHAR RC
LETTERS :CHAR
PR :CHAR
TYPE "½
TYPE :NUM
PR []
SCORE
SCRABBLE
END

TO SCORE
MAKE "VALUE :VALUE + :NUM
MAKE "SCORE :VALUE
SCRABBLE
END

TO START
TYPE "½
TYPE "½
TYPE "½
TYPE "
TYPE [YOUR SCORE IS]
TYPE "½
TYPE :SCORE
MAKE "VALUE 0
MAKE "SCORE 0
WAIT 100
CT
PR [TYPE A WORD. DO NOT TYPE RETURN. TYPE AN * AT THE END
OF THE WORD.]
SCRABBLE
END

MAKE "END "E
MAKE "CHAR "*
MAKE "RHYME "E
MAKE "WORD2 "THRE
MAKE "NEXT "E
MAKE "WORD3 "THR
MAKE "THIRD "R
MAKE "RHYME2 "E
MAKE "NUM 1
MAKE "SCORETWO 0
MAKE "TOTTWO 0
MAKE "SCOREONE 11
MAKE "TOTONE 21
MAKE "VALUE 8
MAKE "SCORE 8
MAKE "PLAYER 1
```

ATARI

```
1 REM ) THIS ROUTINE ALLOWS THE USER T
0
2 REM ) CHANGE THE FILENAME OF NORMAL
3 REM ) DISK FILES. IT READS AND WRITE
S
4 REM ) DIRECTLY FROM & TO THE DISK
5 REM ) DIRECTORY.
6 REM ) 04/13/84                JDM
9 REM
10 KEY=90
20 G.100
40 PRINT:PRINT:PRINT"Potential fatal e
rror":PRINT"Try again":FOR X=1 TO 200:
NEXT X:G.2020
50 PRINT CHR$(125):PRINT:PRINT:PRINT"f
inished....":POKE 752,0:END
70 FL=1:PRINT CHR$9125)
71 PRINT"      "
72 PRINT:PRINT"entry #     filename
   size":PRINT
75 RET.
90 K=0:TRAP 02:INPUT K:TRAP 40000
92 RET.
94 INPUT #3,T$:RET.
99 REM
100 DIM DDIR$(1020),T$(12):DDIR$="  ";
DDIR$(1020)="   ":DDIR$92)=DDIR$
105 OPEN #3,4,0,"E:":POKE 752,1:POS.2,
7:PRINT"a moment please....
110 FOR I=1536 TO 1536+59:READ BYTE:PO
KE I,BYTE:NEXT I:REM SET UP ML ROUTINE
120 PRINT:PRINT:PRINT"Insert disk in D
rive 1":PRINT"and press (RETURN)";:GOS
.KEY
1000 REM
1001 REM ) DISPLAY ROUTINE (
1002 REM
1010 X=USR(1536,ADR(DDIR$)):REM READ D
ISK DIRECTORY
1020 J=0
1050 IF J)63 THEN 50
1055 GOS.70
1060 V=ASC(DDIR$(1+16*J,1+16*J))
1070 IF V=66 OR V=98 THEN PRINT J+1;"
....",DDIR$(6+16*J,13+16*J);".";DDIR$(
14+16*J,16+16*J);"     ";
1080 IF V=66 OR V=98 THEN PRINT ASC(DD
IR$(2+16*J,2+16*J))+256*ASC(DDIR$(3+16
*J,3+16*J))
1085 J=J+1
1090 FL=FL+1:IF FL(17 THEN 1060
1110 POS.2,22:PRINT"Type entry # or (R
ETURN) ";:GOS.KEY
1120 IF K=0 THEN FL=1:G.1050
2000 REM
2001 REM ) CHANGE AND WRITE
```

STRINGER

```
0 GOTO 100
1 REM  STRINGER - a utility to enter
2 REM  String Declaration Statements
3 REM Copyright 1983 - SOFTWARE POLISH
4 REM  Written by Rick Groszkiewicz
5 REM .. Delete Backup File
6 TRAP 8:XIO 33,#1,0,0,"D:STRINGER.BAK"
7 REM .. Rename file to backup
8 TRAP 10:XIO 32,#1,0,0,"D:STRINGER,STRINGER.BAK"
9 REM .. Now save new program
10 SAVE "D:STRINGER":TRAP 40000:STOP
90 REM
95 REM      TITLE SCREEN
100 REM
105 GRAPHICS 17:POKE 712,96:POKE 711,238:POKE 710,60:POKE 709,174:POKE 708,152
110 DL=PEEK(560)+256*PEEK(561)
115 POKE DL+3,71:POKE DL+10,7
120 ? #6;"  SOFTWARE  POLISH"
125 ? #6:? #6;"        presents":? #6:? #6
130 ? #6;"  S T R I N G E R"
135 ? #6:? #6
140 ? #6:? #6;"     A UTILITY"
145 ? #6:? #6;"   FOR  ENTERING"
150 ? #6:? #6;" STRING STATEMENTS"
155 ? #6:? #6:? #6:? #6
160 ? #6:? #6;"press start to begin"
190 REM
195 REM DIMENSION STRINGS/DECLARATIONS
200 REM
205 DIM PMOVE$(44),CURSOR$(11),STRING$(32),PR$(190),MEM$(89),SCREEN$(800),XDIR(15),YDIR(15)
210 POKE 82,1:POKE 83,39
215 OPEN #2,4,0,"K:"
220 PMOVE$="hh"\J+ZF\F^ 1♦*(%_p ♪ ^ Py (%)p\1♦r\ Py♦"
225 CURSOR$="▲?!!!!!!!!?"
230 MEM$="}♦, I, Uh*pNh, Lh, KJpEh, Nh, MJp<h, Uh, T" %KP/J%LP J/0&Up♦ %0p 1K MHPuf LfNJP1$Tp %0p 1K M Pt♦"
235 PM=ADR(PMOVE$):CURSOR=ADR(CURSOR$)
240 SCREEN$=" ":SCREEN$(800)=" ":SCREEN$(2)=SCREEN$
245 IF PEEK(53279)<>6 THEN 245
250 REM P/M GRAPHICS / INITIALIZATION
255 GOSUB 3000
260 REM
490 REM
495 REM MAIN LOOP - MOVE P/M CURSOR
500 REM
505 REM THIS UNSCROLLS THE SCREEN
510 IF PEEK(DMEM+821)<>255 THEN A=USR(ADR(MEM$),ADR(SCREEN$),DMEM+160,800)
515 ST=STICK(0)
520 TR=STRIG(0)
525 REM HORIZONTAL MOVEMENT TESTS
530 X=XDIR(ST)
535 HORZ=HORZ+X
540 IF HORZ<49 THEN HORZ=201
545 IF HORZ>201 THEN HORZ=49
550 REM VERTICAL MOVEMENT TESTS
555 OLDV=VERT
560 Y=YDIR(ST)
565 VERT=VERT+Y
570 IF VERT<79 THEN VERT=191
575 IF VERT>191 THEN VERT=79
580 REM DO P/M MOVEMENT "TOGETHER"
585 POKE HP,HORZ
590 IF Y THEN A=USR(PM,CURSOR,OLDV+P0,VERT+P0)
595 REM GIVE SPEAKER CLICKS WHEN MOVED
600 IF X OR Y THEN POKE 53279,1
605 REM FORMULA TO MOVE GRAPHICS CRSR
610 X=2+(HORZ-53)/4
615 Y=8+(VERT-95)/8
620 POKE 704,PEEK(20):POKE 16,64:POKE 53774,64
625 POSITION X,Y:GET #1,A
630 POSITION X,Y:? CHR$(27);CHR$(A);
635 KEY=0:Z=PEEK(764):IF Z=255 THEN 675
640 REM CHECK KEYBOARD FOR 'DELETE'
645 IF Z=52 THEN GOSUB 2000:GOTO 675
650 REM CHECK FOR LOGO/CAPS/CTRL KEYS
655 IF Z=60 OR Z=124 OR Z=188 THEN POKE 764,255:POKE 702,Z-60:GO TO 675
660 IF Z=39 OR Z=103 OR Z=167 THEN POKE 764,255:POKE 694,128-PEEK(694):GO TO 675
665 GET #2,A:KEY=1
670 REM IF START PRESSED, SAVE STRING
675 IF PEEK(53279)=6 THEN 1000
680 REM IF TRIGGER, PRINT STRING CHAR
685 IF TR AND ( NOT KEY) THEN 500
690 POSITION COL,ROW
695 ? CHR$(27);CHR$(A);
700 COL=COL+1
705 IF COL>39 THEN COL=2:ROW=ROW+1
710 PR$(LEN(PR$)+1)=CHR$(A)
715 FOR I=10 TO 1 STEP -3
720 SOUND 0,A/5+10,10,I
725 NEXT I
730 SOUND 0,0,0,0
735 GOTO 500
1000 REM
1005 REM INSTRUCTIONS TO SAVE STRING
1010 REM
1015 POSITION 1,4
1020 ? "   Are you sure??   (Y/N)"
1025 POKE 764,255
1030 IF PEEK(764)=255 THEN 1030
1035 A=PEEK(764):POKE 764,255
1040 IF A=43 THEN 1060
1045 POSITION 1,4
1050 ? "Press START to save string"
1055 GOTO 500
1060 ? CHR$(125):POKE HP,0
1065 PR$(LEN(PR$)+1)=CHR$(34)
1070 FOR I=1 TO LEN(PR$)
1075 ? CHR$(27);PR$(I,I);:NEXT I
1080 REM CREATE A 'LIST' STATEMENT
1085 LENTH=LEN(STRING$)-1:IF LENTH>3 THEN LENTH=3:POSITION 2,8
1090 ? :? "CLOSE #3:OPEN #3,8,0,";
1095 REM THIS LINE FOR CASSETTE USERS
1100 IF PEEK(128)+256*PEEK(129)<6000 THEN ? CHR$(34);"C:";:GO TO 1115
1105 ? CHR$(34);"D:";
1110 ? STRING$(1,LENTH);NUMBER;".STR";
1115 ? CHR$(34);":PRINT #3;PR$:";
1120 ? " CLOSE #3":? :? :? "RUN":? :?
1125 ? :? "HIT RETURN KEY AND"
1130 ? "THE STRING WILL BE"
1135 ? "SAVED AS SHOWN ABOVE"
1140 ? :? "HIT RETURN AGAIN TO RERUN"
1145 POSITION 0,7
1150 POKE 752,0
1155 END
2000 REM
2005 REM DELETE A CHARACTER FROM STRING
2010 REM
2015 COL=COL-1
2020 IF COL=1 THEN COL=39:ROW=ROW-1
2025 POSITION COL,ROW
2030 ? CHR$(27);CHR$(32);
2035 PR$(LEN(PR$))=""
2040 FOR I=10 TO 255 STEP 40
2045 SOUND 0,I,10,I/17
2050 NEXT I
2055 SOUND 0,0,0,0
2060 POKE 764,255
2065 RETURN
3000 REM
3005 REM INITIALIZATION - P/M GR
3010 REM
3015 Z=4+16*INT(RND(16)*16)
3020 GRAPHICS 0:POKE 712,Z:POKE 710,Z:POKE 709,14
3025 POKE 752,1:?
3030 PMBASE=8*INT(PEEK(561)/8)-8
3035 P0=256*(PMBASE+4)
3040 POKE 704,14+3*16
```

stringer cont

```
3045 HP=53248:POKE HP,0
3050 POKE 53277,3
3055 POKE 54279,PMBASE
3060 POKE 559,62
3065 VERT=135
3070 HORZ=125
3075 A=USR(ADR(MEM$),0,P0,256)
3080 REM
3085 REM SET UP JOYSTICK DIRECTIONS
3090 REM
3095 DATA 0,0, 0,0,  0,0,   0,0, 4,8,4
,-8, 4,0, 0,0, -4,8, -4,-8, -4,0,    0
,0,  0,8, 0,-8, 0,0
3100 RESTORE 3095
3105 FOR I=1 TO 15
3110 READ X,Y
3115 XDIR(I)=X
3120 YDIR(I)=Y
3125 NEXT I
3130 REM
3135 REM INPUT INFO FOR STRING STMT
3140 REM
3145 ? "PLEASE ENTER THE FOLLOWING ITE
MS":POKE 752,0
3150 POSITION 1,3
3155 ? "LINE NUMBER  ?32000"
3160 POSITION 1,5
3165 ? "STRING NAME  ?DUMMY$"
3170 POSITION 1,7
3175 ? "STRING START ?001"
3180 POSITION 1,9
3185 ? "Your result will look like:"
3190 ? :? "32000 DUMMY$(1)=";CHR$(34);
"an unprintable string";CHR$(34)
3195 TRAP 3195
3200 POSITION 14,3:INPUT NUMBER
3205 TRAP 3205
3210 POSITION 14,5:INPUT STRING$
3215 IF STRING$(LEN(STRING$))<>"$" THE
N STRING$(LEN(STRING$)+1)="$"
3220 TRAP 3220
3225 POSITION 14,7:INPUT START
3230 TRAP 40000:POKE 752,1
3235 PR$=STR$(NUMBER)
3240 PR$(LEN(PR$)+1)=" "
3245 PR$(LEN(PR$)+1)=STRING$
3250 PR$(LEN(PR$)+1)="("
3255 PR$(LEN(PR$)+1)=STR$(START)
3260 PR$(LEN(PR$)+1)=")="
3265 PR$(LEN(PR$)+1)=CHR$(34)
3270 POSITION 1,15
3275 ? "Your output will be:"
3280 ? :? PR$;"an unprintable string";
CHR$(34)
3285 ? :? "PRESS ANY KEY TO CONTINUE"
3290 ? " PRESS BREAK KEY TO STOP"
3295 POKE 764,255
3300 IF PEEK(764)=255 THEN 3300
3305 POKE 764,255
4000 REM
4005 REM DRAW THE CHARACTERS ON SCREEN
4010 REM
4015 OPEN #1,12,0,"5:"
4020 POKE 559,62:POKE 752,1
4025 SETCOLOR 1,0,0:SETCOLOR 2,0,10:SE
TCOLOR 4,0,10
4030 POSITION 2,6:FOR I=0 TO 127
4035 REM NEED TO SKIP THE QUOTE (34)
4040 IF I=34 THEN ? "  ";:GOTO 4050
4045 ? CHR$(27);CHR$(I);" ";
4050 NEXT I
4055 FOR I=129 TO 136
4060 ? "  ";:NEXT I
4065 ?
4070 FOR I=0 TO 127
4075 REM NEED TO SKIP THE (CR) (155)
4080 IF I=27 THEN ? "  ";:GOTO 4090
4085 ? CHR$(27);CHR$(I+128);" ";
4090 NEXT I
4095 REM SET UP ROWS/COLUMNS - STRING
4100 ROW=0:COL=LEN(PR$)+2
4105 POKE 82,2:POSITION 2,0
4110 ? PR$
4115 POSITION 1,4
4120 ? "Press START to save string"
4125 REM
4130 REM SETUP TO RESTORE AFTER THE
4135 REM SCREEN INADVERTENTLY SCROLLS
4140 REM
4145 DL=PEEK(560)+256*PEEK(561)
4150 DMEM=PEEK(DL+4)+256*PEEK(DL+5)
4155 A=USR(ADR(MEM$),DMEM+160,ADR(SCRE
EN$),800)
4160 POKE HP,HORZ
4165 A=USR(PM,CURSOR,0,VERT+P0)
4170 RETURN
```

PILOT DIR

```
9990 *DIR T:* *DIR  PILOT Disk Dire
ctory
9991 C:@B710=180
9992 C:$DIR=D:*.*
9993 C:@81373=2
9994 *DIRLOOP  READ:$DIR $FNAME
9995 J(@228=136):*EOF
9996 T:$FNAME  \
9997 J:*DIRLOOP
9998 *EOF  CLOSE:$DIR
9999 E:
```

alterdir cont

```
2002 REM
2020 TRAP 40:PRINT CHR$(125):PRINT"You
 may now alter the filename
2030 PRINT"listed below.  Be sure to k
eep the ":PRINT"proper spacing.":POKE
752,0
2040 POS.2,7:PRINT DDIR$(6+16*(K-1),13
+16*(K-1));")";DDIR$(14+16*(K-1),16+16
*(K-1));:PRINT CHR$(27);:PRINT CHR$(30
)
2050 PRINT"--------  ---":POS.2,7:PRIN
T CHR$(31);:PRINT CHR$(30);:GOS.KEY +4
2055 IF ASC(T$)<65 OR ASC(T$)>90 THEN
40
2060 T$(9T$(10):DDIR$(6+16*(K-1),16+16
*(K-1))=T$
2070 POKE 1543,80:X=USR(1536,ADR(DDIR$
)):G.50
5000 REM
5001 REM ) DATA
5002 REM
5020 D.104,169,1,141,1,3,169,82,141,2,
3,104,141,5,3,104,141,4,3,169,105,141,
10,3,169
5030 D.1,141,11,3,169,8,141,255,6,32,8
3,228,24,173,4,3,105,128,141,4,3,144,3
5035 D.238,5
5040 D.3,238,10,3,206,255,6,208,231,96
```

STAN OCKERS FORKLIFT

```
10 DIM PM$(1):REM *** ENTER THIS LINE
FIRST ***
20 REM *************************
30 REM *       FORKLIFT        *
40 REM *       S. OCKERS       *
50 REM *         10/84         *
60 REM *************************
70 REM ACE Newsletter, 3662 Vine Maple
 Dr.
80 REM Eugene, OR 97405 Nov 84 $12 yea
r
100 GRAPHICS 18:POSITION 7,5:? #6;"Fo
Li":POSITION 5,8:? #6;"INITIALIZING
":POSITION 3,10
110 ? #6;"takes 23 seconds":GOSUB 4000
:POKE 559,0:GOSUB 3000
198 REM
199 REM *** STARTING POSITIONS ***
200 GOSUB 2000:TOPL=C4:YL=TOPL*C8:XL=C
0:SIZL=14:GOSUB 920
210 RP$(C1)=" ":RP$(98)=" ":RP$(C2)=RP
$:YR=16:XR=33:SIZR=C8:GOSUB 930:YR=80:
GOSUB 930:POKE 712,66
220 TOPR=18:TOPC=18:SIZR=C0:SIZC=C0:FR
K=96:LDED=C0:AREA=C2:DIR=RT:X=C0:OSET=
58:BOSET=C6:ROSET=74:WOSET=89
230 CKFLG=C0:FRKC$=FRKL$:BODYC$=BODYL$
:RACKC$=RACKL$:GOSUB 900
298 REM
299 REM *** MAIN LOOP ***
300 STK=STICK(C0)
302 SOUND C1,C0,C0,C8
310 ON AREA GOSUB 400,450,500,520,530,
550,600
390 GOTO 300
398 REM
399 REM *** AREA #1 - LEFT EDGE ***
400 IF STK=UP THEN GOSUB 1050
410 IF (STK=DN) THEN GOSUB 1100
420 IF STK=RT THEN AREA=C2:SOUND C2,60
,12,10:DIR=RT
430 RETURN
448 REM
449 REM *** AREA #2 LEFT PILE ***
450 IF DIR=LT THEN GOSUB 1200
460 IF DIR=RT THEN GOSUB 1250
470 RETURN
498 REM
499 REM *** AREA #3 CENTER ***
500 IF STK=UP THEN GOSUB 1300
502 IF STK=DN THEN GOSUB 1350
504 IF STK=LT THEN AREA=C2:DIR=LT
506 IF STK=RT THEN AREA=C4:DIR=RT
508 RETURN
518 REM
519 REM *** AREA #4 CENTER ***
520 IF DIR=LT THEN GOSUB 1400
522 IF DIR=RT THEN GOSUB 1450
524 RETURN
529 REM *** AREA #5 RIGHT CENTER ***
530 IF STK=UP THEN GOSUB 1300
532 IF STK=DN THEN GOSUB 1350
534 IF STK=LT THEN AREA=C4:DIR=LT
536 IF STK=RT THEN AREA=C6:DIR=RT
538 RETURN
548 REM
549 REM *** AREA #6 RIGHT PILE ***
550 IF SIZR=C6 AND CKFLG=C0 THEN GOSUB
 3500
555 IF DIR=LT THEN GOSUB 1550
560 IF DIR=RT THEN GOSUB 1500
570 RETURN
598 REM
599 REM *** AREA #7 - RIGHT EDGE ***
600 IF STK=UP THEN GOSUB 1650
610 IF STK=DN THEN GOSUB 1700
620 IF STK=LT THEN AREA=C6:DIR=LT
630 RETURN
898 REM
899 REM *** MOVE FORKS & WORDS ***
900 PM$(FRK+523)=FRKC$:POKE 53250,X*C4
+OSET
902 PM$(803)=RACKC$:POKE 53251,X*C4+RO
SET:POKE 53249,X*C4+WOSET
905 A=USR(ADR,X+BOSET,120,BADR,35,C7)
910 IF LDED=C0 THEN RETURN
915 A=USR(ADR,X,Y,CADR,SIZC*C7,C7):RE
TURN
918 REM
919 REM *** REPLACE LEFT PILE ***
920 IF SIZL=C0 THEN RETURN
925 A=USR(ADR,XL,YL,LPADR,SIZL*C7,C7)
:RETURN
928 REM
929 REM *** REPLACE RIGHT PILE ***
930 IF SIZR=C0 THEN RETURN
940 A=USR(ADR,XR,YR,RPADR,SIZR*C7,C7)
:RETURN
998 REM
999 REM *** LOAD FORK LEFT PILE ***
1000 TEMP$=LP$:IF (FRK/C8<TOPL+C2) THE
N RETURN
1005 IF SIZL=C0 THEN RETURN
1010 LDED=C1:C$(C8)=LP$:SIZC=FRK/C8-TO
PL+C2:SIZL=18-FRK/C8:TOPC=TOPL-C1
1020 P=SIZC*C7-C7:C$(P)=B$:IF SIZL)C0
THEN LP$=TEMP$((FRK/C8-TOPL)*C7+C1)
1030 TOPL=FRK/C8:YL=TOPL*C8:POKE 712,7
0:PITCH=50:GOSUB 1800:RETURN
1048 REM
1049 REM *** MOVE UP LEFT SIDE ***
1050 SOUND C1,120,12,C8:IF LDED=C0 THE
N GOSUB 1000
1060 IF (FRK/C8-SIZC)<=C0 THEN RETURN
1070 Y=TOPC*C8:FOR J=C1 TO 16:Y=Y-C1:F
RK=FRK-C1:GOSUB 900:GOSUB 920:IF FRK>1
28 THEN GOSUB 1900
1080 NEXT J:TOPC=TOPC-C2
1090 RETURN
1098 REM
1099 REM *** MOVE DOWN LEFT SIDE ***
1100 SOUND C1,130,12,C8:IF LDED=C0 THE
N RETURN
1110 IF (FRK/C8=TOPL) THEN GOSUB 1150:
RETURN
1120 Y=TOPC*C8:FOR J=C1 TO 16:Y=Y+C1:F
RK=FRK+C1:GOSUB 900:GOSUB 920:IF FRK>1
28 THEN GOSUB 1900
1130 NEXT J:TOPC=TOPC+C2:RETURN
1148 REM
1149 REM *** UNLOAD LEFT SIDE ***
1150 IF LDED=C0 THEN RETURN
1160 TEMP$=C$(C8):IF FRK/C8<18 THEN TE
MP$(SIZC*C7-13)=LP$
1170 LP$=TEMP$:SIZL=SIZL+SIZC-C2:TOPL=
TOPC+C1:YL=TOPL*C8:LDED=C0:POKE 712,66
:PITCH=150:GOSUB 1800:RETURN
1198 REM
1199 REM *** MOVE LEFT AREA #2 ***
1200 SOUND C2,90,12,C6:IF FRK/C8)TOPL
AND LDED=C1 THEN DIR=RT:GOSUB 1850:RET
URN
1205 IF FRK)144 THEN DIR=RT:GOSUB 1850
:RETURN
1210 IF X=C0 THEN AREA=C1:SOUND C2,C0,
C0,C0:RETURN
1220 X=X-C1:GOSUB 900:IF FRK=144 THEN
GOSUB 1900
1230 IF LDED=C1 THEN GOSUB 920
1240 RETURN
1248 REM
1249 REM *** MOVE RIGHT AREA #2 ***
1250 SOUND C2,90,12,C6:IF X=C7 THEN AR
EA=C3:SOUND C2,C0,C0,C0:RETURN
1260 X=X+C1:GOSUB 900
1270 IF LDED=C1 THEN GOSUB 920:IF FRK=
144 THEN GOSUB 1900
1280 RETURN
1298 REM
1299 REM *** GO UP CENTER ***
1300 SOUND C1,120,12,C8:IF (FRK/C8-SIZ
C)<=C0 AND LDED=C1 THEN RETURN
1305 IF FRK<=32 THEN RETURN
1310 FOR J=C1 TO 16:Y=Y-C1:FRK=FRK-C1:
GOSUB 900:NEXT J:TOPC=TOPC-C2:RETURN
1348 REM *** GO DOWN CENTER ***
1350 SOUND C1,130,12,C8:IF (FRK=160) T
HEN RETURN
```

```
1360 FOR J=C1 TO 16:Y=Y+C1:FRK=FRK+C1:
GOSUB 900:NEXT J:TOPC=TOPC+C2:RETURN
1398 REM
1399 REM *** GO LEFT CENTER ***
1400 SOUND C2,90,12,C6:IF X=C6 THEN AR
EA=C3:SOUND C2,C0,C0,C0:RETURN
1403 IF X=23 THEN BODYC$=BLNK$:GOSUB 9
00
1404 IF X=21 THEN WOSET=-40
1405 IF X=17 THEN OSET=58:BOSET=C6:FRK
C$=FRKL$:RACKC$=RACKL$:ROSET=66
1406 IF X=14 THEN ROSET=74
1407 IF X=10 THEN BODYC$=BODYL$:WOSET=
89
1410 X=X-C1:GOSUB 900:RETURN
1448 REM
1449 REM *** GO RIGHT CENTER ***
1450 SOUND C2,90,12,C6:IF X)25 THEN AR
EA=C5:SOUND C2,C0,C0,C0:RETURN
1453 IF X=10 THEN BODYC$=BLNK$:GOSUB 9
00
1454 IF X=11 THEN WOSET=-44:ROSET=64
1455 IF X=15 THEN OSET=49:BOSET=-6:FRK
C$=FRKR$:RACKC$=RACKR$
1456 IF X=17 THEN ROSET=42
1457 IF X=20 THEN BODYC$=BODYR$:WOSET=
26
1460 X=X+C1:GOSUB 900:RETURN
1498 REM
1499 REM *** MOVE RIGHT AREA #6 ***
1500 SOUND C2,90,12,C6:IF FRK/C8)TOPR
AND LDED=C1 THEN DIR=LT:GOSUB 1850:RET
URN
1505 IF FRK)144 THEN DIR=LT:RETURN
1510 IF X=33 THEN AREA=C7:SOUND C2,C0,
C0,C0:RETURN
1520 X=X+C1:GOSUB 900:IF FRK=144 THEN
GOSUB 1900
1530 IF LDED=C1 THEN GOSUB 930
1540 RETURN
1548 REM
1549 REM *** MOVE LEFT AREA #6 ***
1550 SOUND C2,90,12,C6:IF X(26 THEN AR
EA=C5:CKFLG=C0:SOUND C2,C0,C0,C0:RETUR
N
1560 X=X-C1:GOSUB 900:IF FRK=144 THEN
GOSUB 1900
1570 IF LDED=C1 THEN GOSUB 930
1580 RETURN
1599 REM *** LOAD WORDS RIGHT PILE ***
1600 TEMP$=RP$:IF (FRK/C8(TOPR+C2) THE
N RETURN
1605 IF SIZR=C0 THEN RETURN
1610 LDED=C1:C$(C8)=RP$:SIZC=FRK/C8-TO
PR+C2:SIZR=18-FRK/C8:TOPC=TOPR-C1
1620 P=SIZC*C7-C7:C$(P)=B$:IF SIZR)C0
THEN RP$=TEMP$((FRK/C8-TOPR)*C7+C1)
1630 TOPR=FRK/C8:YR=TOPR*C8:POKE 712,7
0:PITCH=50:GOSUB 1800:RETURN
1648 REM
1649 REM *** MOVE UP RIGHT SIDE ***
1650 SOUND C1,120,12,C8:IF LDED=C0 THE
N GOSUB 1680
1660 IF (FRK/C8-SIZC)(=C0 THEN RETURN
1670 Y=TOPC*C8:FOR J=C1 TO 16:Y=Y-C1:F
RK=FRK-C1:GOSUB 900:GOSUB 930:IF FRK)1
28 THEN GOSUB 1900
1680 NEXT J:TOPC=TOPC-C2
1690 RETURN
1698 REM
1699 REM *** MOVE DOWN RIGHT SIDE ***
1700 SOUND C1,130,12,C8:IF LDED=C0 THE
N RETURN
1710 IF (FRK/C8=TOPR) THEN GOSUB 1750:
RETURN
1720 Y=TOPC*C8:FOR J=C1 TO 16:Y=Y+C1:F
RK=FRK+C1:GOSUB 900:GOSUB 930:IF FRK)1
28 THEN GOSUB 1900
1730 NEXT J:TOPC=TOPC+C2:RETURN
1748 REM
1749 REM *** UNLOAD RIGHT SIDE ***
1750 IF LDED=C0 THEN RETURN
1760 TEMP$=C$(C8):IF FRK/C8(18 THEN TE
MP$(SIZC*C7-13)=RP$
1770 RP$=TEMP$:SIZR=SIZR+SIZC-C2:TOPR=
TOPC+C1:YR=TOPR*C8:LDED=C0:POKE 712,66
:PITCH=150:GOSUB 1800:RETURN
1798 REM
1799 REM *** DING ***
1800 FOR J=12 TO C1 STEP -C1:SOUND C0,
PITCH,10,J:NEXT J:SOUND C0,C0,C0,C0:RE
TURN
1848 REM
1849 REM *** CRASH ***
1850 FOR J=15 TO C1 STEP -C1:SOUND C1,
200,C8,J:NEXT J:SOUND C1,C0,C0,C0:RETU
RN
1898 REM
1899 REM *** PRINT THE DOCKS ***
1900 A=USR(ADR,C0,145,DADR,12,C6):A=U
SR(ADR,34,145,DADR,12,C6):RETURN
1998 REM
1999 REM *** SET UP WORDS ***
2000 LP$="  ▄▄▄▄  ▄▄▄▄ ":LP$(98)=" ":
LP$(15)=LP$:RESTORE 2100:PICK=INT(RND(
C0)*C9)+C1
2010 FOR J=C1 TO PICK:READ MTCH$:NEXT
J:CNT=C0:FOR J=C3 TO 87 STEP 14
2020 LINE=INT(RND(C0)*C9)+C1
2021 IF RND(C0)(0.33 THEN LINE=PICK
2022 RESTORE 2100+10*LINE:K=INT(RND(C0
)*C5)+C1:FOR L=C1 TO K:READ WD$:NEXT L
2024 FOR Q=C3 TO J STEP 14:IF WD$=LP$(
Q,Q+C2) THEN 2020
2026 NEXT Q
2030 IF LINE=PICK THEN CNT=CNT+C1:IF C
NT)C3 THEN 2020
2040 IF J=31 AND CNT=C0 THEN 2020
2042 IF J=59 AND CNT=C1 THEN 2020
2044 IF J=87 AND CNT=C2 THEN 2020
2060 LP$(J,J+C2)=WD$:NEXT J
2070 ? CHR$(125):POKE 656,C0:POKE 657,
22:? "get 3 _";MTCH$;" words":POKE 559
,62
2090 RETURN
2100 DATA AN,ED,AT,OT,UN,AD,AG,ID,AP
2110 DATA CAN,PAN,RAN,FAN,MAN
2120 DATA BED,FED,LED,RED,WED
2130 DATA BAT,CAT,SAT,FAT,RAT
2140 DATA HOT,POT,NOT,LOT,ROT
2150 DATA BUN,GUN,FUN,RUN,SUN
2160 DATA MAD,SAD,LAD,BAD,HAD
2170 DATA BAG,SAG,TAG,RAG,WAG
2180 DATA BID,LID,HID,DID,RID
2190 DATA CAP,LAP,SAP,NAP,ZAP
2998 REM
2999 REM *** INITIALIZE ***
3000 C0=0:C1=1:C2=2:C3=3:C4=4:C5=5:C6=
6:C7=7:C8=8:C9=9
3010 DIM LP$(98),RP$(98),C$(112),TEMP$
(98),W$(C7),B$(C7),F$(C9)
3020 LPADR=ADR(LP$):RPADR=ADR(RP$):CAD
R=ADR(C$):TADR=ADR(TEMP$):FADR=ADR(F$)
:C$(1)=" ":C$(112)=" ":C$(2)=C$:B$=C$
3030 UP=14:DN=13:LT=11:RT=C7
3050 GRAPHICS C8:POKE 559,C0:PM=PEEK(1
06)-40:POKE 54279,PM:POKE 53258,C1:POK
E 706,166:POKE 53277,C3
3060 VVTP=PEEK(134)+PEEK(135)*256:STAR
P=ADR(PM$):OFF5=PM*256+1024-STARP:HI=I
NT(OFF5/256):LO=OFF5-HI*256
3070 POKE VVTP+C2,LO:POKE VVTP+C3,HI:P
OKE VVTP+C5,C4:POKE VVTP+C6,C0:POKE VV
TP+C7,C4
3075 PM$(C1)=CHR$(C0):PM$(1024)=CHR$(C
0):PM$(C2)=PM$
3080 DIM FRKL$(20),FRKR$(20),FRKC$(20)
:FOR J=C2 TO 16:FRKL$(J)=CHR$(C1):FRKR
$(J)=CHR$(128):NEXT J
3090 FOR J=17 TO 19:FRKL$(J)=CHR$(255)
:FRKR$(J)=CHR$(255):NEXT J:FRKL$(20)=C
HR$(C0):FRKR$(20)=CHR$(C0)
3100 POKE 710,32:POKE 623,C1:DIM MTCH$
(C2),WD$(C3)
3110 DIM BODYL$(35),BLNK$(35):BODYL$="
  ▌ ▄▄▌ ▐▄▄▌ ▄▄▄▄ ▼ "
3120 DIM BODYR$(35),BODYC$(35):BODYR$=
" ▌  ▐▄▄  ▐▄▄▌ ▄▄▄▄ ▼ ":
```

OCKERS FORKLIFT

```
BADR=ADR(BODYC$)
3130 BLNK$(C1)=" ":BLNK$(35)=" ":BLNK$
(C2)=BLNK$:FOR J=34 TO 194:PM$(J,J)=CH
R$(255):NEXT J
3140 POKE (704),42:POKE 53248,110:POKE
 53256,C3
3150 DIM RACKL$(160),RACKR$(160),RACKC
$(160):RACKL$(C1)=CHR$(192):RACKL$(C2)
=CHR$(64):RACKL$(150)=" "
3160 RACKL$(C3)=RACKL$:RACKR$(C1)=CHR$
(C3):RACKR$(C2)=CHR$(C2):RACKR$(150)="
 ":RACKR$(C3)=RACKR$
3170 RESTORE 3180:FOR J=150 TO 160:REA
D A:RACKL$(J,J)=CHR$(A):NEXT J
3172 RESTORE 3182:FOR J=150 TO 160:REA
D A:RACKR$(J,J)=CHR$(A):NEXT J
3180 DATA 8,28,62,54,99,99,99,54,62,28
,8
3182 DATA 16,56,124,108,198,198,198,10
8,124,56,16
3190 POKE 707,C4
3200 RESTORE 3210:FOR J=444 TO 450:REA
D A:PM$(J,J)=CHR$(A):NEXT J:POKE 705,C
4
3210 DATA 8,28,62,62,62,28,8
3220 DIM DOCK$(12):DOCK$="[redacted]||||||
":DADR=ADR(DOCK$):GOSUB 1900
3230 A=PEEK(560)+256*PEEK(561)
3240 IF PEEK(A)<>66 THEN A=A+C1:GOTO 3
240
3250 POKE A,70:POKE A+C3,C6:POKE A+C4,
C6:POKE A+C5,C6
3300 RETURN
3498 REM
3499 REM *** CHECK FOR DONE ***
3500 CNT=C0:CKFLG=C1:PITCH=40
3510 IF RP$(C4,C5)=MTCH$ AND RP$(18,19
)=MTCH$ AND RP$(32,33)=MTCH$ THEN 3550
3520 RETURN
3550 FOR K=C0 TO C9:POKE 656,C0:POKE 6
57,22:? "   great job!!!   ":PITCH=100
-C5*K:GOSUB 1800
3560 POKE 656,C0:POKE 657,22:? "
          ":FOR L=C0 TO 10:NEXT L:NE
XT K
3580 POKE 559,C0:BODYC$=BLNK$:A=USR(AA
DR,X+BOSET,120,BADR,35,C7):PM$(FRK+523
,FRK+542)=PM$(FRK+443,FRK+562)
3590 POP :GOTO 200
3998 REM
3999 REM *** GR. 8 ROUTINE ***
4000 DIM A$(257):RESTORE 4010:FOR J=1
TO 257:READ A:A$(J,J)=CHR$(A):NEXT J:A
ADR=ADR(A$):RETURN
4010 DATA 216,104,104,104,133,203,104,
104,133,204,169,0,133,205,6,204,38,205
,6,204,38,205,6,204,38
4020 DATA 205,165,204,24,101,88,133,20
6,165,205,101,89,133,207,6,204,38,205,
6,204,38,205,165,204,24
4030 DATA 101,206,133,206,165,205,101,
207,133,207,165,206,24,101,203,133,206
,141,133,6,165,207,105,0,133
4040 DATA 207,141,134,6,104,133,213,10
4,133,212,104,104,141,129,6,206,129,6,
104,104,141,131,6
4050 DATA 169,0,141,132,6,141,130,6,16
9,0,141,128,6,172,130,6,177,212,16,5,2
06,128,6,41,127,201,32
4060 DATA 176,5,24,105,64,16,7,201,96,
176,3,56,233,32,133,204,169,0,133,205,
133,208,6,204,38
4070 DATA 205,6,204,38,205,6,204,38,20
5,165,205,24,109,244,2,133,205,164,208
,177,204,77,128,6,172
4075 DATA 132,6,145,206
4080 DATA 230,208,165,208,201,8,240,15
,165,206,24,105,40,133,206,144,227,230
,207,208,223
4090 DATA 144,160,238,132,6,238,130,6,
206,129,6,48,43,173,132,6,205,131,6,20
8,22,169,0,141,132,6,24,173,133,6
4100 DATA 105,64,141,133,6,173,134,6,1
05,1,141,134,6,173,133,6,133,206,173,1
34,6,133,207,24,144,200,96
```

OCKERS BIBLIOGRAPHY

PERIODICAL BIBLIOGRAPHY
MICROS AND THE DISABLED A.N.A.L.O.G #18,April '84,'Communication for the Handicapped' by Michael Long (p15) An Atari scanning program using a pressure activated 'Puff and Sip' switch. BYTE The Sept. '82 issue of Byte was devoted to computers and the disabled. Among the articles were: 'Computers Can Play a Dual Role for Disabled Individuals' by Gregg Vanderheiden (p136) A summary of ways computers can be used by the disabled, pointing out that they should not only be thought of as providing special functions, but should give disabled individuals access to standard software. 'A New Horizon for Nonvocal Communication Devices' by Patrick Demasco & Ricard Foulds (p166) Describes using a Panasonic portable computer with Votrax Type-'N-Talk as a scanning communicator. Also describes using a digitizing tablet to select letter clusters to create messages. 'Minspeak' by Bruce Baker (p186) A system is outlined where a large number of complete sentences can be choosen for reproduction using a very small number of physical responses (keystroke on a specially designed keyboard). 'FDA Regulation of computerized Medical Devices' by Jorgens, Bruch & Huston (p204) What designers of medically related hardware and software should know. 'Talking Terminals' by David Stoffel (p218) Terminals are available which speak what is sent to them, (for about $5000). This article describes three of these. 'Braille Writing in Pascal by Alfred Fant Jr. (p250) Using a latex cushion of the platten, a regular printer can be used to print Braille. 'Adaptive-Firmware Card for the Apple II' by Paul Schwejda & Gregg Vanderheiden (p276) A hardware solution that allows disabled individuals to run standard, unmodified software.'LOGO: An Approach to Educating Disbled Children' by Weir, Russel & Valente (p342) Creating action-oriented learning environments and putting pupils in charge of their own learning. 'Review of The Cognivox V10-1003' by Dr. William Murray (p231) A voice recognition unit for the Apple II can be trained to recognize a set of up to 32 words or short phrases (cost-$295). 'The Abilityphone (review)' by William Rush (p240) A security device for the disabled can provide emergency dialing, monitoring and repeated dialing of a phone number. 'Build the Microvox Text-To-Speech Synthesizer' by Steve Ciarcia Contains its own microprocessor and text to speech algorithm to change RS232 supplied text directly into speech (part II in Oct. '82 issue). BYTE Nov. '83 'Build The H-Com Handicapped Communicator' by Steve Ciarcia (p36) Describes construction of a scanner based on a Intel 8748 single chip microcomputer. COMPUTE! A series 'Micros With The Handicapped' by Marshall Curtis and Susan Semancik featured articles on creating an electronics communication board in Basic. Among the topics discussed were: April '82 (p98) Outline of series covering 5 fundamental areas June '82 (p94) consideration in setting up the menu,Oct. '82 (p124) a digression- The game 'MasterMind' with single button input Jan. '83 (p124) entry of data into menu Apr. '83 (p135) saving menus on disk or tape and inputing directly to the screen June '83 (p164) making selections from menus Nov. '83 (p233) a complete program for the VIC20 with 8K COMPUTE! Feb. '84 'Special Education Applications' by Semancik & Benvenuti An Atari counting program with joytick input for special education students. COMPUTE Sept. '84 'Aids for The Blind' (p122) by Glenn Kleiman A review of ways computers can help people who have impaired vision or are blind (include technological aids reference). CREATIVE COMPUTING Mar. '82 'Interactive Programming' (p146) Hardware and software techniques for the physically limited. CREATIVE COMPUTING Aug. '83 'Handicapped and Working' by Kirby Morgan (p142) First hand account of how computers can help the handicapped to work at home. CREATIVE COMPUTING Oct. '83 'Communicating in Code' by Wolfger Schneider (p222) A Morse code generating switch acts as a keyboard replacement. CREATIVE COMPUTING The 'Computing For the Handicapped' column has been taken over by Shel Talmy and includes the following installments: Dec. '83 (p332) Describes the Osborne 1 base AVOS system Feb. '84 (p222) Describes the VIM Voice Input Module Sept. '84 Various hardware and software aids including VOCA, S.A.M.,KolaPad and a voice based recipe file program. IEEE MICRO June '83 'An Integrated Workstation for the Visually Handicapped' by Grossner, Radhakrishnan & Pospiech (p8) CP/M based system with Intex 'Talker' and homemade Braille 'display' unit. IEEE MICRO June '83 'Applying Anticipatory Text Selection in a Writing Aid for People with Severe Motor Impairment' by Hekathorne & Childress (p17) A text processor with scanning input tries to speed up the selection process by anticipating what is coming up from what was last entered. IEEE MICRO Aug. '83 'The Mod Keyboard' by Nelson, Korba, Park & Crabtree (p7) Uses a VIC20 with ROM and battery backed RAM as a keyboard replacement for another computer. IEEE MICRO Aug. '83 'A Microcomputer-based Laboratory Aid for Visually Impaired Students' by Lunney et. al. (p19) CP/M based system for data acquisition and analysis has voice output.

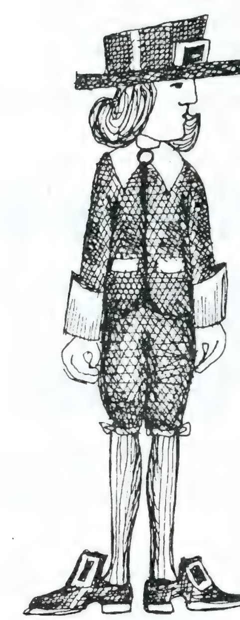
MEETING
WEDNESDAY
NOV 14th
7:30pm
SOUTH
EUGENE
HIGH

GREG MENKE disk File disassembler

```
10 REM Memory/Disk file disassembler
11 REM
12 REM By Greg Menke
13 REM
14 REM 9/9/84 V2.0
15 REM
16 REM
17 REM
19 REM HEX and DECimal conversion
20 REM routines by Shane Rolin.
22 REM
24 REM See the October 1983 ACE.
30 REM
35 REM
60 GRAPHICS 0:SETCOLOR 2,0,0
70 DIM A$(1020),OUT$(20),I$(20),IN$(20),HEX$(4),H$(23),MNEM$(4),X$(1),OP$(20),T$(20),ADDR$(10)
71 SIZE=FRE(0)-1024:DIM BUFF$(SIZE)
72 BUFF$(1)="♥":BUFF$(SIZE)="♥":BUFF$(2)=BUFF$
75 H$="0123456789ABCDEF"
80 A$(1)="♥":A$(1020)="♥":A$(2)=A$
83 POKE 752,1:? "%Initializing ... "
90 READ A,T$:IF A)-1 THEN A$(A*4-3,A*4)=T$:POSITION 19,0:? T$(1,3);:GOTO 90
95 POKE 752,0
100 ? "%Disassemble from Disk or Memory ?";
102 CLOSE #3:OPEN #3,4,0,"K":GET #3,A:? CHR$(A):CLOSE #3:I$=CHR$(A)
105 IF I$()"D" THEN 120
107 ? "%":CLOSE #1:OPEN #1,6,0,"D:*.*":TRAP 115
110 INPUT #1;T$:? T$,:INPUT #1;T$:? T$:GOTO 110
115 ? :? :? INT(SIZE/125);" free sectors of file space."
118 CLOSE #1:? :? "Enter BINARY LOAD filename.":? :INPUT IN$:IF IN$="" THEN 100
120 ? "%Disassemble to Screen, Disk file, or Printer ?";
125 CLOSE #3:OPEN #3,4,0,"K":GET #3,A:? CHR$(A):CLOSE #3:T$=CHR$(A)
130 IF T$="S" OR T$=CHR$(155) THEN OUT$="E:"
140 IF T$="P" THEN OUT$="P:"
150 IF T$="D" THEN ? "%Enter filename ";:INPUT OUT$:IF OUT$="" THEN 120
160 REM
165 REM
170 REM
180 IF I$="D" THEN 190
182 ? "%Enter start address in SHEX or DEC.":? :INPUT T$:IF T$="" THEN 120
183 IF T$(1,1)="$" THEN HEX$=T$(2,LEN(T$)):GOSUB 4000:START=NUM:GOTO 186
185 START=VAL(T$)
186 ND=65535:GOTO 300
187 REM
188 REM
189 REM
190 OPEN #1,4,0,IN$
200 GET #1,X:GET #1,Y:GET #1,A:GET #1,B:START=A+B*256
210 GET #1,A:GET #1,B:ND=A+B*256
220 IF X=255 AND Y=255 THEN 260
230 ? :? :? IN$;" is NOT a binary load file!!":? :? :? "Press START";:CLOSE #1
240 IF PEEK(53279)<>6 THEN 240
250 GOTO 100
260 ? "%":H=INT(ADR(BUFF$)/256):L=ADR(BUFF$)-(256*H):IOCB=834+1*16:POKE IOCB,7:POKE IOCB+2,L:POKE IOCB+3,H
261 POKE IOCB+6,255:POKE IOCB+7,255
265 A=USR(ADR("hhh[LV[":1*16):CLOSE #1
283 REM
284 REM
290 REM
300 OPEN #2,8,0,OUT$:SETCOLOR 2,0,0:LOOP=400:LOC=START
310 CONVERT=3000:OFF=1
315 ? "%Press any key to abort....":? :POKE 764,255
320 REM
330 REM
400 IF I$="D" THEN A=ASC(BUFF$(OFF,OFF)):OFF=OFF+1:GOTO 420
410 A=PEEK(LOC)
420 ADD=1:MNEM=A:NUM=LOC:GOSUB CONVERT:ADD=0
425 IF PEEK(764)<255 THEN GOSUB 700
430 ADDR$="$":ADDR$(2)=HEX$:ADDR$(LEN(ADDR$)+1)="  "
440 IF A=0 THEN FL=1:MNEM$="BRK":GOTO 600
450 FL=0:MNEM$=A$(MNEM*4-3,MNEM*4)
460 X$=MNEM$(4,4):MNEM$(4)=""
470 IF X$="1" THEN OP$="A":GOTO 600
480 IF X$="2" THEN GOSUB 1000:OP$="#$":OP$(3)=HEX$
490 IF X$="3" THEN GOSUB 1000:OP$="$":OP$(2)=HEX$
500 IF X$="4" OR X$="5" THEN GOSUB 1000:OP$="$":OP$(2)=HEX$:GOSUB 1010+(10*X$="5")
510 IF X$<"6" OR X$)"8" THEN 520
514 GOSUB 1000:OP$="$":OP$(4,5)=HEX$:GOSUB 1000:OP$(2,3)=HEX$
515 IF X$)"6" THEN GOSUB 1010+(10*X$="8")
520 IF X$="9" THEN OP$=""
525 IF X$="A" THEN GOSUB 1500
530 IF X$<"B" OR X$)"D" THEN 540
533 GOSUB 1000:OP$="($":OP$(3,4)=HEX$:IF X$="B" THEN GOSUB 1010
535 OP$(LEN(OP$)+1)=")":IF X$="C" THEN GOSUB 1020
540 IF X$="♥" THEN OP$="-$":NUM=MNEM:GOSUB CONVERT:OP$(LEN(OP$)+1)=HEX$:OP$(LEN(OP$)+1)="-":MNEM$="???"
591 REM
592 REM
593 REM
600 ? #2;ADDR$;MNEM$;:IF FL=0 THEN ? #2;" ";OP$;
610 ? #2:LOC=LOC+1
620 IF LOC)ND THEN ? :? "%End reached.":? :? "Listing from ";START;" to ";ND;".":? :CLOSE #2:CLOSE #1:END
630 GOTO LOOP
700 ? :? :? "START = Abort.":? "SELECT = Continue.":? "OPTION = End.":? :?
710 POKE 764,255:IF PEEK(53279)=6 THEN CLOSE #2:CLOSE #1:POP :GOTO 100
720 IF PEEK(53279)=5 THEN RETURN
730 IF PEEK(53279)<>3 THEN 710
740 POP :CLOSE #2:CLOSE #1:? :END
1000 LOC=LOC+1
1002 IF LOC)ND THEN ? :? "%End reached.":? :? "Listing from ";START;" to ";ND;".":? :CLOSE #2:CLOSE #1:END
1005 NUM=PEEK(LOC):IF I$="D" THEN NUM=ASC(BUFF$(OFF,OFF)):OFF=OFF+1
1007 GOSUB CONVERT:A=NUM:RETURN
1010 OP$(LEN(OP$)+1)=",X":RETURN
1020 OP$(LEN(OP$)+1)=",Y":RETURN
1500 GOSUB 1000:N=A
1510 IF A)127 THEN A=LOC-(255-N)
1520 IF A<128 THEN A=LOC+N+1
1530 NUM=A:GOSUB CONVERT
1540 OP$="$":OP$(2)=HEX$:RETURN
1999 REM
2000 DATA 105,ADC2,101,ADC3,117,ADC4,109,ADC6,125,ADC7,121,ADC8,97,ADCB,113,ADCC,41,AND2,37,AND3,53,AND4
2010 DATA 45,AND6,61,AND7,57,AND8,33,AND8,49,ANDC,10,ASL1,6,ASL3,22,ASL4,14,ASL6,30,ASL7
2020 DATA 144,BCCA,176,BCSA,240,BEQA,36,BIT3,44,BIT6,48,BMIA,208,BNEA,16,BPLA,80,BVCA,112,BVSA
2030 DATA 24,CLC9,216,CLD9,88,CLI9,184,CLV9,201,CMP2,197,CMP3,213,CMP4,205,CMP6,221,CMP7,217,CMP8,193,CMP8
```

MENKE CONT

```
2040 DATA 209,CMPC,224,CPX2,228,CPX3,2
36,CPX6,192,CPY2,196,CPY3,204,CPY6,198
,DEC3,214,DEC4,206,DEC6,222,DEC7
2050 DATA 202,DEX9,136,DEY9,73,EOR2,69
,EOR3,85,EOR4,77,EOR6,93,EOR7,89,EOR8,
65,EORB,81,EORC
2060 DATA 230,INC3,246,INC4,238,INC6,2
54,INC7,232,INX9,200,INY9,76,JMP6,108,
JMPD,32,JSR6
2070 DATA 169,LDA2,165,LDA3,181,LDA4,1
73,LDA6,189,LDA7,185,LDA8,161,LDAB,177
,LDAC,162,LDX2,166,LDX3
2080 DATA 182,LDX5,174,LDX6,190,LDX8,1
60,LDY2,164,LDY3,180,LDY4,172,LDY6,188
,LDY7,74,LSR1
2090 DATA 70,LSR3,86,LSR4,78,LSR6,94,L
SR7,234,NOP9,9,ORA2,5,ORA3,21,ORA4,13,
ORA6,29,ORA7,25,ORA8,1,ORAB
2100 DATA 17,ORAC,72,PHA9,8,PHP9,104,P
LA9,40,PLP9,42,ROL1,38,ROL3,54,ROL4,46
,ROL6,62,ROL7
2110 DATA 106,ROR1,102,ROR3,118,ROR4,1
10,ROR6,126,ROR7,64,RTI9,96,RTS9,233,S
BC2,229,SBC3
2120 DATA 245,SBC4,237,SBC6,253,SBC7,2
49,SBC8,225,SBCB,241,SBCC,56,SEC9,248,
SED9,120,SEI9
2130 DATA 133,STA3,149,STA4,141,STA6,1
57,STA7,153,STA8,129,STAB,145,STAC,134
,STX3,150,STX5,142,STX6
2140 DATA 132,STY3,148,STY4,140,STY6,1
70,TAX9,168,TAY9,186,TSX9,138,TXA9,154
,TXS9,152,TYA9,-1,NULL
3000 REM CONVERT DEC TO HEX
3020 M=USR(ADR("hh U♥h T♥ H T♥), U♥nT♥
NU♥nT♥NU♥nT♥NU♥nT♥┴i0I:0 |i/IG+/ J♥PP♦
"),NUM)
3030 FOR G=203 TO 206:HEX$(G-202)=CHR$
(PEEK(G)):NEXT G:IF HEX$(1,2)="00" AND
 ADD=0 THEN HEX$=HEX$(3,4)
3040 RETURN
4000 REM CONVERT HEX TO DEC
4010 A=0:FOR I=1 TO LEN(HEX$):IF (HEX$
(I,I)<"A" OR HEX$(I,I)>"F") AND (HEX$(
I,I)<"0" OR HEX$(I,I)>"9") THEN A=1
4011 NEXT I:IF A THEN NUM=0:? "5555Inv
alid hexidecimal parameter!!!":RETURN
4020 AD=ADR(HEX$):L=LEN(HEX$)
4030 NUM=USR(ADR("hh,T,h,Lhh,N ♥,T U1LI
001I@0 TA0)IG+%┴i ,J,/T&U/T&U/T&U/T&U┴e
I,T%Ui♥,UHDNPI♦"),AD,L)
4040 RETURN
BADR=ADR(BODYC$)
3130 BLNK$(C1)=" ":BLNK$(35)=" ":BLNK$
(C2)=BLNK$:FOR J=34 TO 194:PM$(J,J)=CH
R$(255):NEXT J
3140 POKE (704),42:POKE 53248,110:POKE
 53256,C3
3150 DIM RACKL$(160),RACKR$(160),RACKC
$(160):RACKL$(C1)=CHR$(192):RACKL$(C2)
=CHR$(64):RACKL$(150)=" "
3160 RACKL$(C3)=RACKL$:RACKR$(C1)=CHR$
(C3):RACKR$(C2)=CHR$(C2):RACKR$(150)="
 ":RACKR$(C3)=RACKR$
3170 RESTORE 3180:FOR J=150 TO 160:REA
D A:RACKL$(J,J)=CHR$(A):NEXT J
3172 RESTORE 3182:FOR J=150 TO 160:REA
D A:RACKR$(J,J)=CHR$(A):NEXT J
3180 DATA 8,28,62,54,99,99,99,54,62,28
,8
3182 DATA 16,56,124,108,198,198,198,10
8,124,56,16
3190 POKE 707,C4
3200 RESTORE 3210:FOR J=444 TO 450:REA
D A:PM$(J,J)=CHR$(A):NEXT J:POKE 705,C
4
3210 DATA 8,28,62,62,62,28,8
3220 DIM DOCK$(12):DOCK$="      ||||||
":DADR=ADR(DOCK$):GOSUB 1900
3230 A=PEEK(560)+256*PEEK(561)
3240 IF PEEK(A)<>66 THEN A=A+C1:GOTO 3
240
3250 POKE A,70:POKE A+C3,C6:POKE A+C4,
C6:POKE A+C5,C6
3300 RETURN
3498 REM
3499 REM *** CHECK FOR DONE ***
3500 CNT=C0:CKFLG=C1:PITCH=40
3510 IF RP$(C4,C5)=MTCH$ AND RP$(18,19
)=MTCH$ AND RP$(32,33)=MTCH$ THEN 3550
3520 RETURN
3550 FOR K=C0 TO C9:POKE 656,C0:POKE 6
57,22:? "   great job!!!   ":PITCH=100
-C5*K:GOSUB 1800
3560 POKE 656,C0:POKE 657,22:? "
           ":FOR L=C0 TO 10:NEXT L:NE
XT K
3580 POKE 559,C0:BODYC$=BLNK$:A=USR(AA
DR,X+BOSET,120,BADR,35,C7):PM$(FRK+523
,FRK+542)=PM$(FRK+443,FRK+562)
3590 POP :GOTO 200
3998 REM
3999 REM *** GR. 8 ROUTINE ***
4000 DIM A$(257):RESTORE 4010:FOR J=1
TO 257:READ A:A$(J,J)=CHR$(A):NEXT J:A
ADR=ADR(A$):RETURN
4010 DATA 216,104,104,104,133,203,104,
104,133,204,169,0,133,205,6,204,38,205
,6,204,38,205,6,204,38
4020 DATA 205,165,204,24,101,88,133,20
6,165,205,101,89,133,207,6,204,38,205,
6,204,38,205,165,204,24
4030 DATA 101,206,133,206,165,205,101,
207,133,207,165,206,24,101,203,133,206
,141,133,6,165,207,105,0,133
4040 DATA 207,141,134,6,104,133,213,10
4,133,212,104,104,141,129,6,206,129,6,
104,104,141,131,6
4050 DATA 169,0,141,132,6,141,130,6,16
9,0,141,128,6,172,130,6,177,212,16,5,2
06,128,6,41,127,201,32
4060 DATA 176,5,24,105,64,16,7,201,96,
176,3,56,233,32,133,204,169,0,133,205,
133,208,6,204,38
4070 DATA 205,6,204,38,205,6,204,38,20
5,165,205,24,109,244,2,133,205,164,208
,177,204,77,128,6,172
4075 DATA 132,6,145,206
4080 DATA 230,208,165,208,201,8,240,15
,165,206,24,105,40,133,206,144,227,230
,207,208,223
4090 DATA 144,160,238,132,6,238,130,6,
206,129,6,48,43,173,132,6,205,131,6,20
8,22,169,0,141,132,6,24,173,133,6
4100 DATA 105,64,141,133,6,173,134,6,1
05,1,141,134,6,173,133,6,133,206,173,1
34,6,133,207,24,144,200,96
```



STRINGER

String Entry Program

Recently, numerous articles have appeared in **ANTIC** which contain string declaration statements consisting of numerous control characters. The advantage of direct string declarations is they use less memory than comparable READ/DATA statements. In addition, the amount of time required for program initialization is significantly reduced.

There is a major drawback to direct string declaration statements — they require much more effort by the person who keys in the program! It is hard to read the program, interpret the control characters shown, and type in the combination of keystrokes to get the correct characters on the screen.

This is why I wrote STRINGER. The program should simplify the entry of unusual strings. The program uses Player/Missile graphics for a cursor which moves over the entire ATASCII character set on the screen. The cursor is joystick controlled, and a press of the fire button places the selected character in a string declaration statement at the top of the screen. Characters may be entered directly from the keyboard. [DELETE] can be used to delete characters from the string declaration statement.

KEYING IN THE PROGRAM

The STRINGER program in Listing one is entirely self contained. It can be keyed in and RUN immediately if you desire to do so. Disk drive owners should note the routine in lines 5-10. This routine eases the task of entering the programs while avoiding 'Atari lock-up'. By typing G.5 in direct mode, the program will delete the backup file, rename the current version of the program to the backup file, and SAVE the new version to disk for you. This is an efficient way to recycle file names on the disk. It avoids guesswork as to what file name was used to save the last version of the program.

Listing two contains two subroutines which can be used to speed up the initialization of the STRINGER program. The program in Listing two is optional, and it does not have to be entered to use the STRINGER program. Listing three contains the Assembly Language source code for the machine language routine used in the STRINGER program. The source code is shown for those of you who are interested. It does not have to be keyed in to use the STRINGER program.

PROGRAM USAGE

After the title screen comes up, the program initializes for a few seconds. Press [START] when prompted to do so. The program will ask you to enter the line number, the name of the string variable, and the starting index in the string.

The main part of the program executes next. First, the program places the character set at the bottom of the screen. Next, the string statement is placed at the top of the screen. Then, you can move a cursor across the screen via the joystick. When the cursor is positioned over the desired character, a press of the fire button will place the character in the string statement at the top of the screen. You may enter characters directly from the keyboard instead of using the joystick. If you incorrectly enter a character into the string statement, press [DELETE] to delete the character.

After you finish creating the string statement, press [START]. After verifying you are ready to save the string statement, the program will place a line of BASIC code on the screen, and the cursor will be placed on that line. Simply press [RETURN] to LIST the string to disk or cassette. If you have another string statement to create, you can press [RETURN] to RUN the program again.

INITIALIZATION SPEED-UP

When RUN, the program takes a few seconds to read the 89 data items and place them in the MEM$ string. It is possible to avoid this delay using the subroutines in Listing two. These lines of BASIC should be keyed in and LISTed so they can be ENTERed into the STRINGER program.

Be sure the STRINGER program has been SAVEd, and it is operating correctly. Press [SYSTEM RESET] to re-enable the BREAK key. Run the program, and when prompted to press [START], press [BREAK] instead. Next, type in "GO TO 6000" in direct mode, and press [RETURN].

The subroutine at line 6000 will create the string declaration statement for MEM$, and place it into the body of the program. In addition, a few other lines of the program will be changed. The subroutine at line 6100 will print the correct LIST/NEW/ENTER/SAVE lines on the screen for you. By simply pressing [RETURN] several times, you can create a new version of STRINGER containing a string declaration statement. The program will be shorter than the original, and it will initialize in only a second!

MACHINE LANGUAGE ROUTINE

The MEM$ string contains a dual purpose machine language routine. The Assembly Language source code for this routine is shown in Listing three. There are two calling sequences from BASIC. When called via:

A = USR(ADR(MEM$),0,ADDRESS,NUMBER)

the routine will fill memory starting at ADDRESS with zeroes for the specified NUMBER of bytes. This is used at line 3075 to clear the memory used for the P/M graphics.

The alternate calling sequence is:

A = USR(ADR(MEM$),SOURCE,DEST,NUMBER)

The routine will copy the specified NUMBER of bytes starting at the SOURCE address in memory to the DEST address in memory. This routine is used to copy 800 bytes of screen memory in lines 510 and 4255. It is also used to move ten bytes of pP/M data (stored in CURSOR$) at lines 590 and 4165. (WARNING: this routine is not 'smart' enough to handle cases where overlapping area of memory are copied!)

Another use for this routine is to copy the entire character set from ROM to RAM. If you had defined the variable CHSET as the location in memory for the character set, the statement

A = USR(ADR(MEM$),57344,CHSET,1024)

would be at least 15 seconds faster than using a FOR/NEXT loop to do the same thing!

TECHNICAL NOTES

Lines 3080-3125 set up two arrays of directional values for joystick movement. Instead of reading the joystick and testing various values to determine the direction, one line can do it:

ST = STICK(X):X = XDIR(ST):Y = YDIR(ST).

This handy trick first appeared in the M.A.C.E. Journal in a doodling program in 1982.

Lines 655 and 660 contain a routine to check whether [CAPS/LOWER] or [Atari Logo] has been pressed. This is necessary because, without these two lines, input from the keyboard does not work well. The reason is the GET statement at line 665 does not recognize these two special keys as "real" characters from the keyboard. The program simply stops looping when either of these keys are pressed, since the GET function is waiting for another keypress. Now you may press either of these keys, and the program does not get confused. Credit for the hint regarding the special memory locations used in lines 655 and 660 goes to Russ Hickman from Birmingham, Alabama. Thanks, Russ!

Lines 4145 to 4155 and 510 eliminated a problem involving the screen editor. When a character is placed in the rightmost column of the screen, the screen editor will **sometimes** scroll the lines below that point. After this happened a few times, part of the character set could be lost off the bottom of the screen!

The solution is to save the entire screen beyond the first four lines, which is where the string declaration statement is placed. The screen is saved at lines 4145-4155. If the screen does scroll, it is detected at line 510, and the USR call restores the screen quickly. The "jerking" of the screen when this happens is barely noticeable.

PROGRAM OUTLINE

The STRINGER program is organized in an unusual way. The program flow is as follows:

Line 0 jumps to line 90 to avoid the backup routine.
Lines 90 through 230 produce a title screen.
Line 235 jumps to Line 5000.
Lines 5000 through 5250 execute and return to line 235.
Lines 240 through 250 execute.
Line 255 jumps to line 3000.
Lines 3000 through 4170 execute and return to 255.
Lines 490 through 735 execute as the main program loop.

An outline of the functions of the program modules is as follows:

Line Numbers	Description
5-10	Backup routine for keying in program
90-155	Title screen
190-255	Initialization and setup
490-735	Main program loop
1000-1155	Create 'LIST' statement for string
2000-2065	Delete character from string
3000-3075	Set up P/M graphics
3080-3125	Set up joystick directions
3130-3305	Set up line number and string name
4000-4120	Place character set on the screen
4125-4170	Save entire screen
5000-5250	Set up machine language routine

This structure is convoluted, but there are some reasons for this setup. The main program loop has been placed at the beginning of the program for speed of execution. Also, if the subroutines in Listing two are ENTERed into the STRINGER program and executed, all of the lines after 5000 can be eliminated. The first step of the initialization was placed at the end of the program in anticipation of being eliminated!

Description of Variables

A	Temporary Hold Variable
CURSOR	Address of CURSOR$ string
CURSOR$	Holds data for P/M graphics cursor
COL	Current column location of regular cursor
DISK	Flag: 1 if DOS has been loaded, 0 if not
DL	Location of display list in memory
DMEM	Location of display data for screen
HORZ	Horizontal position of P/M graphics cursor
HP	Memory position pointer for player zero
I	Miscellaneous loop counter
KEY	Value of keypress from GET function
LENTH	Length of STRING$
MEM	Address of MEM$ string
MEM$	String which holds machine language routine
NUMBER	Line number for string declaration statement
OLDV	Old vertical position of P/M graphics cursor
P0	Memory location for start of data for player zero
PMBASE	Page pointer for P/M base address
PR$	Holds string declaration statement from program
ROW	Current row for regular cursor
SCREEN$	String which holds twenty lines of screen data
ST	Value of leftmost joystick
START	Starting index of string variable in program
STRING$	Holds name of string variable in string statement
TR	Value of trigger of leftmost joystick
VERT	Current vertical position of P/M graphics cursor
X	Used for horizontal cursor movement
XDIR(15)	Horizontal movement data for joystick
Y	Used for vertical cursor movement
YDIR(15)	Vertical movement data for joystick
Z	Holds value of keypress from PEEK(764)

SUMMARY

The STRINGER program is a useful utility for entering string declaration statements. In addition, this program illustrates how to use several tricks, as well as a machine language routine, to do its work more efficiently. Feel free to use these routines in your own programs.

— Rick Groszkiewicz

835/1030 MODEMS

(reprint: Aug-Sep 1984 S.P.A.C.E.)

The following information was obtained on the Atari SIG on CompuServe. It is a condensation of the documentation for a public domain handler for the 835/1030 Atari direct connect modems. The handler is available on the N.W. Micronet, (206) 535-2837. When used as an AUTORUN.SYS it will allow the use of programs such as AMODEM with these modems.

The difference between modems is the way they handle the OPEN and CLOSE statements. Because of timing requirements, an RS232 device must be turned off if you want to use other peripherals like disk drives and printers. 850-based terminal programs do this by executing a CLOSE channel command before executing any I/O to other devices, and then an OPEN channel when they are ready once again for terminal I/O.

Unfortunately, a CLOSE channel issued to an 835/1030 modem will shut it off, hanging up the phone in the process. There are special commands to turn off an 835/1030 modem without disconnecting it — "Suspend" and "Resume". In an 835/1030 terminal program, you only OPEN the modem once, at the beginning of the program. Suspend and Resume commands are used to stop the modem temporarily to use other devices.

835/1030 Specific Commands

There are 9 commands specific to the 835/1030 modem. They are all passed without error, if an 850 is being used (with the exception of Suspend and Resume commands which are translated to their 850 equivalents). T: may need to be substituted for R:, depending upon the handler used.

XIO 73, #channel,0,0,"R:"
Puts the 835/1030 in Originate mode.

XIO 74, #channel,0,0,"R:"
Puts the 835/1030 in Answer mode.

XIO 75, #channel,ASC("#"),0,"R:"
Dials one digit. The "#"represents an ASCII character from 0 to 9.

XIO 76, #channel,0,0,"R:"
Takes the modem "off hook". Picture this as taking hold of a telephone receiver and lifting it off the hook. Thus, off hook means putting the modem "on" the telephone line.

XIO 77, #channel,0,0,"R:"
Puts the modem on hook. The opposite of off hook — hangs up the phone.

XIO 82, #channel,0,0,"R:"
835 modems only: Routes the telephone signal to your TV or monitor speaker. Useful to listen in on a dialing sequence to hear whether or not the other side answers.

XIO 83, #channel,0,0,"R:"
835 modems only: Turns off the TV or monitor speaker.

XIO 89, #channel,0,0,"R:"
Resume command. See above.

XIO 90, #channel,0,0,"R:"
Suspend command. See above.

Some Usage Points

1. Before you can execute most 835/1030 commands, you must Resume the modem if it has been Suspended.

2. You should take the modem off hook before beginning a dialing procedure.

3. When using an 835/1030 modem, you can detect when the carrier has been dropped. Issue a Status command, and then check memory location 746 ($2EA). If the value is greater than 127 (high bit is on) then the carrier is present. Bit 0 of the same location is the hook status (a 1 means modem is off hook).

— Tom Neitzel

65C02

Optimized Systems has available a CMOS version of the 6502 microprocessor, called a 65C02, which can directly replace the normal 6502 in an Atari computer. This chip requires less power, therby running a lot cooler, and for the assembly language programmer, it offers 27 new OP codes. Currently, however, MAC/65 is the only assembler to support these codes unless you write your own Macros, or include it via a .BYTE command. Here are the new commands with a brief description of each:

BRA —branch always. This instruction works like all the other branch instructions, except it always branchs, and is therefor like a JMP but taking up one less byte of memory and one less cycle to execute.

INA & DEA. Increment and decrement the Accumulator. Works the same as INX, DEX, INY, & DEY.

PHX, PHY, PLX, PLY. These instructions work like the PLA and PHA instructions, only pushing the respective register instead of the Accumulator

STZ. This stores a zero into the following location but doesn't affect any register. Address modes available are ABSOLUTE; ABS,X; (ZERO PAGE); (ZPG,X)

TRB. This complements the Accumulator, AND's it with the specified memory location, and stores the results in the meory location.

TSB. This ORA's the Accumulator with the memory location, and stores the results in the memory location. Both TRB & TSB use only ABSOLUE and ZEROPAGE addressing

JMP (ABSOLUTE,X). This instruction takes the absolute address, adds the X register, and jumps to that location. It is a very powerful way of setting up a table of JMP addresses, which are then indexed through the X register.

Additionaly the BIT instruction has two new addressing modes: ABSOLUTE,X and ZEROPAGE,X.

There is also a very useful new addressing mode. A common assembly language instruction sequence is: LDY #0
LDA (zero page),Y

The new addressing mode is OPCODE (zero page). The Y register is not used, but one can use the indirect mode as if Y was set to zero. The following instructions can be used with this mode: ADC, AND, CMP, EOR, LDA, ORA, SBC, STA.

A Pilgrimage to Mecca

This past week I attended a seminar in the San Francisco area for two days. Since I don't get down to that region very often I took the opportunity to plan several meetings while I was there. There was much to do, so I scheduled 4 days in the Bay area.

My first stop was a visit with our friends at ANTIC magazine. You may be happy to hear ANTIC will be placing a stronger emphasis on USER GROUPS in the future. They will be working closely in this effort with MICROBITS and OSS (Optimized System Software) in providing meaningful support to the User Groups. ANTIC will also be making a special effort to keep you all abreast of the current news from or about ATARI. (More about ATARI later). I spent a very pleasant afternoon with Jim Capparel and Gary Yost, discussing the User Group issue as well as technical goodies such as HARD DISK DRIVES, EXPANSION CHASSIS, 80 COLUMN CARDS and RAMDISKS for the XL computers. Gary and I had a very pleasant lunch at THE WATERFRONT, where the prawns are outrageously delicious. Before I left ANTIC, Gary presented me with a copy of "THE BEST OF ANTIC volume 1". In some of my slower moments I have perused this fine volume and wish I could give you a review, but am a bit busy as yet. I will probably hand this to a TRUSTED member of ACE long enough for them to do a proper review next month.

The next day I visited OSS. Bill Wilkinson and Mark Rose were kind enough to devote the afternoon to discussions of exactly which features are important to have in a DOS for a PARALLEL HARD DRIVE. Bill also demonstrated the new WRITER'S TOOL word processor.

AT LAST, somebody has a really powerful, DOS compatible word processor. (I have always used LETTER PERFECT by LJK, and getting my articles ready for submission has been an excercise in self abuse, since I normally transmit them over the MODEM.) The WRITER'S TOOL has NEARLY ALL of the capabilities of LETTER PERFECT(tm) and is user-friendly to boot. Don't get me wrong, I think Letter Perfect is still the most powerful word processor available for the Atari, and will continue to use it for my personal files, but OSS has another winner on their hands.

Friday I left the seminar a bit early in order to meet with BRIAN KERR and JOHN SKRUCH, marketing managers of Hardware and Software respectively. I went into the meeting with some reservations about what I might hear and see, and was very pleasantly surprised. Brian and John are both enthusiastic and knowledgeable. They are extremely aware of the circumstances resulting in the virtual death of ATARI, Inc. and are determined to avoid those pitfalls. There is a new spirit at ATARI. Indeed, there is a new ATARI at ATARI. Take heart: the venerable 800 XL will continue to be produced in quantities, and at prices which will probably induce people such as myself to go out and buy an extra or two. (spares, you know.)

The peripheral products the old ATARI released with their equipment are being evaluated, and those not measuring up will be dropped, while the company will pick up lines of TOP QUALITY peripherals to replace them with. NEW and exciting software is being selected for future release. Negotiations are under way with various suppliers, vendors marketing types and such.

I expect the January CES to be a banner event for ATARI. I have reason to believe they will have some SHOW STOPPERS!! Overall, I think the purchase of ATARI by the Tramiel family is probably going to be the best thing to happen to Atari since the lamentable take-over by WARNER COMMUNICATIONS. After all, If Jack Tramiel can take a second-rate turkey of a computer like the COMMODORE 64 and make it the NUMBER 1 SELLER among home computers, think what he can do with the help of his sons and a FIRST RATE computer like the ATARI.

ATARI USER GROUPS will not be abandoned. Many of you are aware the old ATARI spent a small fortune on "support" in the past. The support amounted to very little in the way of worthwhile or legitimate assistance to the user groups. While the new ATARI is not able to spend large amounts of cash on this, a plan has been developed to allow the USER GROUPS, 3RD PARTY VENDORS, and ATARI CORP. to mutually help each other to the fullest extent at a minimum of cost. ANTIC, OSS, MICROBITS, and CHAOS were heavily involved in the development of the new system. You will undoubtedly hear more of this in the next few months.

I came away from San Francisco with a deep sense of relief at the attitudes and willingness to commit time and resources displayed among those I spoke with. Peace be upon you, my children; all is well in the land of ATARI.

— Kirt E. Stockwell
Roving past president

VP's RAMBLINGS

We have started the new service called BUG BUSTERS. The first people to be a part of this new group are listed elsewhere in this newsletter. While all but Greg Menke are from the Eugene area we hope more members from other states or even countries will volunteer to help with this project.

There isn't too much on the Atari scene right now. But by Christmas I think we will see quite abit happening telling us of things to come. Although we hear of things nothing concrete has shown itself at this point. Let us hope that what we hear will come to pass.

The BBS is working just fine and while we are getting quite afew callers we are not hearing from our members on how you like the board and also how about uploading some of the software you developed so we can share it with the other callers. Try it, the feed back you get might either improve your programs or prompt you to go on and do more.

Next month I'll tell you about some of the software that costs alot but doesn't do much but take your money.

— LARRY GOLD

EASING HANDICAPS WITH A COMPUTER

Our school serves the mentally and physically handicapped residents of Ashtabula County Ohio and provides individual instruction to its students. These children are moderately to severly mentally handicapped and/or developmentally disabled (for example Cerebral Palsey).

Atari computers were chosen for their ease of use, low cost, and good graphics. These factors continue in importance since commercial software for the handicapped is limited and expensive. A good collection of adaptable programs have been obtained from user groups.

Few publishers of textbooks or software, care to be concerned with the needs of a relatively small market. As teachers, we must meet our student's needs. With the computer these needs can be met and our efforts shared!

Public domain software currently represents over 80% of our courseware. Low cost and ease of modification have made this possible. Our purpose for modifing (or writing) programs is:

1.Start at appropriate levels. It has to be possible to be fun.
2.Advance or increase at the right rate (get harder in smaller steps).
3.Give accurate scoring and level of accomplishment.
4.Error trap input statements (make it easier and clearer to use).

Computers have enriched the learning process. The PILOT language has been used to author programs and as a easy learning language for our students. Our use of computers is classroom based and all teachers have access to mobil computer work stations.

As a special education teacher I have spent a lot of time making up materials (such as worksheets) to conform with my students' abilities. Although the computer may not be easier to use, once a program is entered it is far easier to copy, so appropriate lessons may be shared (or changed) at little or no expense.

Atari is weak in customer service. It lacks the profitability to offer extensive support. Yet it is now low enough in cost to make it useful for helping an individual student in the classroom. But successful use may require some information from users who are willing to share their knowledge and experience. This can be an advantage: without the profit motive of a sales comission the micro-computer user can be guided by more impartial evaluations of programs and equipment. My classroom's micro has the position of an individual assistant who always understands the child, reinforces (immediately) the correct responses, and is enjoyable and encouraging.

I want to hear from other special users of the Atari. Our programs range from pre-school to about 6th grade level. I will be happy to supply information about our software and copies of some public domain programs.

Carl Schwartz
Ashtabula Co. Bd. of MSPR/DD
2505 South Ridge East
Ashtabula, Ohio,44004

LIBRARY LIST

ATARI COMPUTER ENTHUSIASTS EXCHANGE LIBRARY(rev. 4/15/84)
The ACE Exchange Library is a service designed to allow the sharing of user-written programs. There are two ways to acquire programs from the ACE exchange library: By ordering the disk(s) you desire and enclosing the appropriate amount of money, or by sending us programs you have written and wish to share with others, in exchange for your choice of any disk in the library. We strongly support this second way of obtaining programs, and will go out of our way to help you share your programs with other users. Please see the back page for complete details.

Disks avaliable for double or single density drives. Please specify double density with order.
We have had SOME success with saving programs to tape. If you wish to take a chance, indicate your choice is TAPE.(Tape option not available for disks marked with *).

GAMES DISK #1, $8
ALIEN--Shoot down alien ships--simple graphics.
BLKJCK--a fairly good game of blackjack.
BOMBERS--Simple game; shoot bombs as they fall.
CONCEN--the game of concentration.
DARKSTAR--maneuver your ship to destroy darkstar.
GALLERY--Good Missile-Player graphics tutorial.
HORSERAC--bet on your favorites and lose your bankroll.
MATCHES--a variation on nim.
MISSILE--protect cities from missiles (40k required).
MOBSTERS--Shoot before they get you!
SIMON--repeat after me; great for parties.
SMASH--all-text car racing game.
SPY--find the spies hidden in 10 x 10 grid.
TOWERS--of Hanoi; good thinker-puzzle.
TTT--tic-tac-toe game.

GAMES DISK #2, $8 *
Contains ADVENTURE, by MAX. This disk is weeks of adventure by itself. The program allows you to write your own adventure, with the extended file generation option.

GAMES DISK #3, $8
APTITUDE--A cute quiz; test your programming aptitude.
CIVILWAR--Will the South or the North rise again. Text only.
CLEWSO--Detective game, who murdered the host?
DOGBITE--Deliver mail without getting bit. Text.
ELEC--Makes decisions affecting service. Good simulation. Text.
LANDER--ACE lander, by Stan Ockers. Good one!

DEMOS DISK #2, $8
APPLEATR--Atari worm eats bytes out of an apple.
BOTCH--The newest in acronym-laden language.
BOOKLIST--Library inventory system; Good program!
CARDS--Graphics for a card game; no game logic.
DIALOG--Computer/psychologist conversion.
HORSE--Graphics; horse running across screen.
HORSES--R-rated horse running across screen.
HORSEV--Variable speed horse demo.
NITEMARE--The day your computer took over!
STAR--Plots star w/colors in Gr. 7
STARWARS.MUS--From the movie, arranged by Aaron Ness.
XMASTREE--Seasonal; good music selection for Christmas.
VEGAS--Slot machine, by Matt Loveless.
SCROLL--Video scroll all directions! (requires 40k).

UTILITIES DISK #1, $8
ALTUSE9--Demonstrates writing/reading an entire screen.
AUTORUN.CAS--Autorun boot for your cassette programs.
BUBLSORT--Demonstrates simple, slow sorting algorythm.
CLOAD.AUT--Automatic cload routine, for your program.
DATASTMT.GEN--Generates data statements for memory.
DAYOWEEK--Subroutine determines day of week, given date.
DECIDUMP--Memory dump in decimal.
DISASM--Analyze memory with 6502 assembler instructions.
DISKTAPE--Dump Diskfiles to tape; great for backup, exchange.
ERRTRAP--Error trapper, describes errors.
HELP--Read messages, documentation placed on disk by,
HELP.WRT--Writes text to disk member to generate documentation, instructions, etc.
HEXADEC--Convert hex to decimal.
HOTSTUFF--Multiple purpose directory-printing menu.
LABELDSK--Generates small directories to fit on small disks.
LISTER--To list programs in column-widths using Atari 825.
MODEM--ACE's Downloader modem program, with improvements.
NOTEIN & NOTEOUT--Demonstration of NOTE and POINT.
PEEKER--Looks at memory at specified locations.
PMGDEMO--Superfast player-missile graphics demo.
RENUM--One of the best renumbering program we've seen, donated by MICROBITS!
SCREENPR--Screen to print for 825.
SUPRCOMP--Compares two similar Basic programs with list of differences.

The documentation available for many of the above programs is weak at best. We have attempted to code instructions into REMs at the beginning of the more difficult programs.

DEMOS DISK #3,
FUGUE2-Music anyone?
OCTADRAW-Creative, reflective shapes.

STARWARP--All-text starwars adventure program.
TREASURE--Map-search for treasure with clues. Stick graphics.

GAMES DISK #4, $8
ROBOTWAR--Can you escape?? Two levels of play.
STARDSTR--Shoot the stars.
LANDER--Great lunar lander.
STOPSOUN--Good at sound recognition.
PRICE--Guess the prices.
MISSILE--Protect your cities from missile attack.
ALIEN--More cities to protect. Four levels of play.
DOGGIES--Got your thinking cap on. A puzzle from Stan Ockers.

DEMOS DISK #1, $8
BACKWARD--Demos use of poke to modify screen to print upside down.
BOW--Graphics plot of 3-leaf bow; good trigonometry.
BOXDEMO--draws nested colored boxes.
CHOPSTIX--single-tone music demo.
CIRCLES--Draws circles; experiment to learn plotting!
CLOCK.DIG--Own your very own computer-controlled digital clock!
ELLIPSE-Draws ellipses.
ETCHSKCH--Draw on screen; good technique for joystick movement.
ERTHQWAK--Shakes house; or is it the TV?
GIGGLE--Electronic light show, from Interface Age.
JAZZ--Multiple voice computer-generated Jazz.
KEYBOARD--Piano simulation.
LOGO--128 colored Atari logo on screen.
MAGIC--Pick a letter; the computer will tell you which one.
MARQUEE--Put your name in lights, or leave a banner on your Atari.
MOVEABOX--Move a box around screen
OXYGENE--Computer-generated random sounds.
PUFF-The magic dragon; music (the dragon's invisible).
ROCKET--Demo using print stmts and sound.
SHADING--3-dimensional box changes colors using keyboard & setcolor.

BARGRAPH-Display data in bar graph form.
STAR-Graphic display.
DIATONIC-Sounds.
MESSAGE-Use your Atari for a message display.
STARSHIP-The Enterprise, in full color.
INVENTORY-Home inventory-using cassette files.
STRINGAR-Computer generated display.
MICROTXT-Text editor in basic.
SOUNDSTK-Experiment with sound using the joystick.
ADSHOW-Put your message on a moving marque. No size limit.
POEM-Atari generated poetry.

UTILITIES DISK #2, $8
EXAMINE-Look at any Sector on Disk.
LPOIR-MX-80 printer utility.
EPSHAND.DEM-MX 80 printer demo.
MX80.IN1-Test your printer cable.
TVTEST-Test patterns for your T.V.
AUTORUN.BLD-Use this program to make your own AUTORUN files.
VARIABLE.LST-List variables in Basic programs.
PLOT825-Use your 825 printer as a Plotter/Printer.
UPDNLOAD-Data Transfer from the 850 interface.
QUESTION.FMT-Random question/statement retrevial.
TIMECLOK-Real time clock runs as you program/play.

BEST OF MACE, $8
GRAPH--Demonstration of the 9 text & graphic modes.
RUNWAY--Landing simulation/game; difficult but fairly good.
SLIDE--Competitive creative addition game.
PRICE--The price is right; guess the price: computer gives hints.
CALCNT--Calorie counter--totals calories for a variety of foods.
ENEMY--Get them before they get you; good graphics action.
STRING.CRE--Create data statements from strings in memory.
CHU--Demonstration of VERY FAST missile-player graphics.
SEARCH--A takeoff on the treasure matrix game.
NUMBER.LI--A number-progression educational game.
MULT--Multiplication drills.

SCOPY 810, $15
ACE is licensed to sell this high quality-fast duplicating software. SCOPY will copy a full disk in two passes with 48K memory. Multiple copies are even faster. Excellent for club libraries, etc. Complete documentation provided. SCOPY is not public domain.

Atari Computer Enthusiasts

A.C.E. is an independent, non-profit and tax exempt computer club and user's group with no connection to the Atari Company, a division of Warner Communication Company. We are a group interested in educating our members in the use of the Atari Computer and in giving the latest News, Reviews and Rumors.

All our articles, reviews and programs come from you, our members.

Our membership is world-wide; membership fees include the A.C.E. Newsletter. Dues are $12 a year for U.S., and $22 a year Overseas Airmail and include about 10 issues a year of the ACE Newsletter.
Subscription Dep't: 3662 Vine Maple Dr., Eugene, OR 97405.

* President— Robert Browning, 90 W. Myoak Dr., Eugene, OR 97404 503-689-1513
Vice Pres— Larry Gold, 1927 McLean Blvd., Eugene, OR 97405 503-686-1490
Secretary— Bruce Ebling, 1501 River Loop #1, Eugene, OR 97404 503-688-6872
Librarian— Ron and Aaron Ness, 374 Blackfoot, Eugene, OR 97404 503—689-7106.
Editors—
Mike Dunn, 3662 Vine Maple Dr., Eugene, OR 97405 / 503-344-6193
Jim Bumpas, 4405 Dillard Rd., Eugene, OR 97405 503 / 484-9925
E.R.A.C.E. (Education SIG Editor)—
Ali Erickson, 295 Foxtail Dr., Eugene, OR 97405 / 503-687-1133
E.R.A.C.E. Corresponding Secretary—
Robert Browning, 90 W. Myoak, Eugene, OR 97404 / 503-689-1513

Send 50c stamps or coin ($1 overseas) to the Ness' for the new, updated ACE Library List-new in May 84 !

Best of ACE books

Volume 1 contains bound issues of the ACE Newsletter from the first issue, Oct 81 to June of 1982

Volume 2 covers July 1982 to June 1983

Only $12 each ($2 extra for Airmail). Available only from:

George Suetsugu
45-602 Apuapu St
Kanoehe, HI 96744

SortFinder 1.2
A composite index of Atari related articles from 5 popular computer periodicals from Apr '81 to June '83, including ACE. Only $6 for ACE member from:
Jim Carr, Valley Soft
2660 S.W. DeArmond
Corvallis, OR 97333

ATARI COMPUTER ENTHUSIASTS
3662 Vine Maple Dr. Eugene OR 97405

FIRST CLASS MAIL

Bulletin Board
(503) 343-4352

On line 24 hours a day, except for servicing and updating. Consists of a Tara equipped 48K Atari 400 with a TARA keyboard, 2 double-density double sided disk drives with an ATR 8000 interface, 2- 8" double density disk drives, an Epson MX80 printer, a Hayes SmartModem; running the ARMUDIC Bulletin Board software written by Frank L. Huband, 1206 N. Stafford St., Arlington, VA 22201. See the Nov '82 issue for complete details.